

UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

**FORTTRAN Programs For
Properties of Water and Steam,
and Solubilities of Solid Silica Phases**

Laura A. Keith¹

Paul T. Delaney¹

Open-File Report 84-690

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards and stratigraphic nomenclature.

¹U.S. Geological Survey
2255 North Gemini Drive
Flagstaff, Arizona 86001

1984

CONTENTS

Page

Abstract	III
Introduction	1
Equations	1
Program Notes	3
H2O	3
H2ODR	3
H201	6
H202	6
SATH2O	7
SATDR	7
SATTP	7
SATROW	7
SATROS	8
SILSOL	8
SILICA	8
TSIO2	9
User-Supplied FORTRAN Programs	9
Example of Use of the Main Programs	10
References	20
Program Listings	21
Main Programs	
H2O	21
SATH2O	24
SILSOL	27
Subroutines for Equation of State for H ₂ O	
H2ODR	35
H201	41
H202	43
SATDR	47
SATTP	49
SATROW	50
SATROS	51

Subroutines for Solubilities of Solid SiO_2 Phases

SILICA	52
TSIO2	56

FIGURES

Page

Relations of main programs and subroutines	4
--	---

TABLES

Page

Symbols Used	5
Arguments for Option Parameter in Main Program H2O	6

ABSTRACT

FORTTRAN programs are presented to calculate thermodynamic properties and viscosity of pure H_2O , including coexisting water and steam, solubilities of solid SiO_2 phases as functions of temperature and pressure, and equilibrium temperatures as functions of aqueous-silica concentrations. The thermodynamic relations are from Keenan et al. (1978); dynamic-viscosity equations are from Watson et al. (1980). Reynolds (1979) provides a root-solving procedure for the equation of state of water such that any two properties among temperature, density, pressure, enthalpy or entropy can be entered as known parameters. Equations for the solubility of quartz, amorphous silica, chalcedony, α -cristobalite and β -cristobalite in dilute water are calculated following Fournier (1981), Fournier and Potter (1982a, b) and Fournier (1984). In addition, kinetic precipitation-rate constants are calculated using equations of Rimstidt and Barnes (1980). Water properties are expressed in units of meters, kilograms and seconds, and temperature in degrees Celsius. Concentration units are molal, molar and parts per million of SiO_2 , and kilograms of H_4SiO_4 per cubic meter of water.

INTRODUCTION

Knowledge of the properties of water over broad temperature and pressure ranges, as well as the solubilities of solid SiO_2 phases, are frequently used by geologists and geochemists. This report presents FORTRAN programs which calculate such properties and which are easy, efficient and flexible to use. The main programs are executed interactively; however, the computational sub-routines can be called from any user-supplied program. First, we summarize important features of the equations, and the temperature and pressure ranges where they are applicable. We then describe the individual programs and present a sample interactive terminal session to illustrate their use. The programs are listed at the end of this report.

EQUATIONS

The programs incorporate the most complete thermodynamic and viscosity data of water and solubilities of silica phases presently available. Thermodynamic relations of water including the liquid, vapor and solid phases within the ranges of temperature and pressure

$$0^\circ\text{C} < t < 1300^\circ\text{C} , \quad 0 \text{ MPa} < p < 100 \text{ MPa} \quad (\text{equation of state})$$

as presented in Keenan et al. (1978). (Note that 100 MPa = 1000 bars.) The equations for the dynamic viscosity of water and steam are from Watson et al. (1980) and are applicable within the ranges:

$$\begin{array}{ll} 0^\circ\text{C} < t < 150^\circ\text{C} , & 0 \text{ MPa} < p < 500 \text{ MPa} \\ 150^\circ\text{C} < t < 600^\circ\text{C} , & 0 \text{ MPa} < p < 350 \text{ MPa} \\ 600^\circ\text{C} < t < 900^\circ\text{C} , & 0 \text{ MPa} < p < 300 \text{ MPa.} \end{array} \quad (\text{viscosity equation})$$

All equations for solubilities of solid silica phases are appropriate only for dilute concentrations of aqueous silica in water. The equation of quartz solubility is from Fournier and Potter (1982a) and is accurate within the ranges

$$25^\circ\text{C} < t < 900^\circ\text{C} , \quad 0 \text{ MPa} < p < 1,000 \text{ MPa.} \quad (\text{quartz solubility})$$

For coexisting water and steam, the equation provides accurate solubilities for the steam phase only above 250°C. If pressures are low or at saturation, Fournier and Potter (1982b) present an equation for saturation temperature as a function of the quartz solubility for the liquid water. The solubility equation for amorphous silica was derived by Fournier (1984) by interpolation of equations (4) and (5) in Fournier and Marshall (1983) and is given by

$$\log m_{as} = 0.4278 - 1229.36 T^{-1} + 88870 T^{-2} + 1153.3 T^{-1} \log v \\ - 0.012143 T \log v + 1.2213 \times 10^5 T^2 \log v$$

where m_{as} is molality of dissolved silica, T is absolute temperature (degrees Kelvin), and v is the specific volume of water. (All symbols used in this report are given in Table 1.) This equation can be applied to water within the ranges

$$20^\circ\text{C} < t < 340^\circ\text{C}, \quad 0 \text{ MPa} < p < 100 \text{ MPa.} \quad (\text{am-sil solubility})$$

and for a coexisting steam phase above 250°C. For conditions of liquid water coexisting with steam, Fournier (1981) presents a solubility equation for amorphous silica which is expressed as a function of saturation temperature. Existing solubility equations for chalcedony, α -cristobalite and β -cristobalite are wholly temperature dependent, and are restricted to the range

$$0^\circ\text{C} < t < 250^\circ\text{C} \quad (\text{chalcedony, } \alpha\text{- and } \beta\text{-cristobalite solubility})$$

(Fournier, 1981). The kinetic precipitation-rate equation of Rimstidt and Barnes (1980) for the reaction between solid phases of SiO_2 and water or steam is a function of temperature only and is valid within the ranges

$$0^\circ\text{C} < t < 300^\circ\text{C}, \quad 0 \text{ MPa} < p < 50 \text{ MPa.} \quad (\text{rate equation})$$

The user is encouraged to consult these sources for additional details regarding the derivation, use and limitations of these equations.

PROGRAM NOTES

The programs are written in FORTRAN 77, which has "structured branching options" (see Meissener and Organick, 1980, p. 480) not available in previous implementations of the FORTRAN language. Relations among the main programs and subroutines are shown in Figure 1. For example, when assembling SATH20 to calculate the properties of coexisting water and steam, subroutines SATDR, SATTP, SATROV, SATROL, H2ODR, H201 and H202 must all be linked. The user should refer to Figure 1 when assembling the programs. To ease interpretation of the FORTRAN code, we fully document the programs and employ notation that resembles parameters and variables in the published equations.

The following is a description of the main programs and the subroutines that they call. Argument listings of subroutines are given. (Table 1 provides a list of symbols). Main programs are identified as MP in these descriptions; arguments are entered interactively by the user after receiving prompts from the program.

H20 (MP) This main program allows the user to specify any two known parameters among temperature, density, pressure, enthalpy and entropy and, if necessary, initial guesses of temperature and/or density by defining an integer variable NOP, as illustrated in Table 2. After calling the subroutine H2ODR, H20 displays the properties for pure water that are most commonly of interest: temperature, pressure, density, thermal expansion coefficient, compressibility, dynamic viscosity, heat capacity at constant volume, heat capacity at constant pressure, enthalpy, internal energy, entropy, Helmholtz free energy, and Gibbs free energy.

H2ODR (input: some combination of t , p , ρ , S or H , and NOP, ERR; output: t , p , ρ , S , H , α , β , ψ , U , C_v , C_p , G , μ) All parameters passed to and from H2ODR are single-precision floating-point numbers, with the exception of NOP and ERR which are integers. The equation of state of water is written is given as a function of temperature and density. Reynolds (1979) presents this driver program which uses a Newton-Raphson type root-solving technique so that pressure, enthalpy or entropy may also be entered as known parameters. H2ODR calls the equation-of-state subroutines H201 and H202, supplying successive

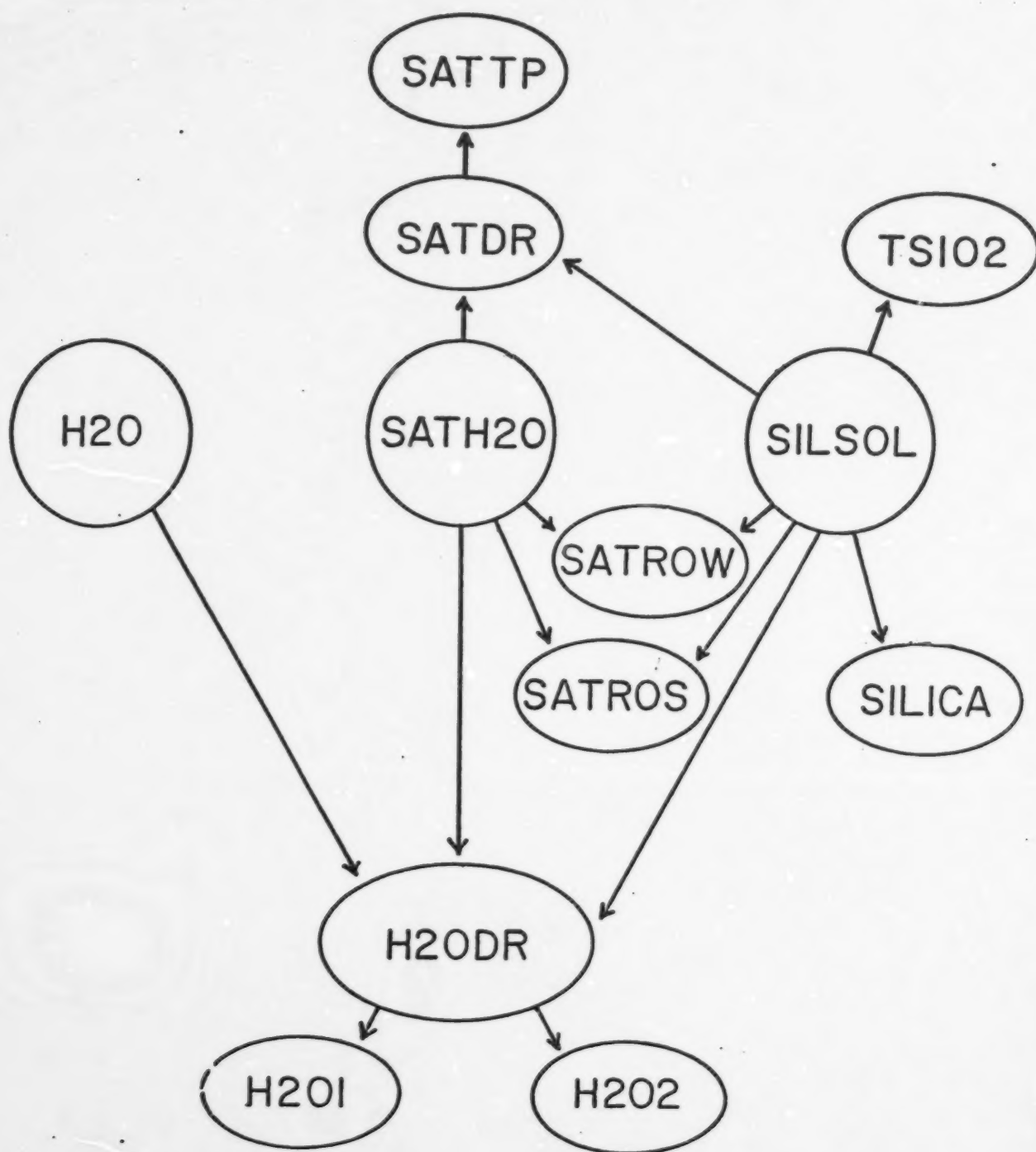


Figure 1. Relations of main programs (circles) to subroutines (ellipses).

Arrows point from the calling program to the called subroutine.

Table 1. Symbols Used in Descriptions of Subroutines.

c	concentration of SiO_2 in solution
C_p	specific heat capacity at constant pressure, $\text{J/kg}^\circ\text{C}$
C_v	specific heat capacity at constant volume, $\text{J/kg}^\circ\text{C}$
G	specific Gibbs free energy, J/kg
H	specific enthalpy, J/kg
k_-	kinetic precipitation-rate constant, m/s and $\text{kg/m}^2\text{s}$
NOP	integer variable to identify known parameters in H2O programs
NOS	integer variable to identify known parameters in SAT programs
NS	integer variable to identify presence of coexisting water and steam
p	pressure, Pa
s	solubility of SiO_2 phase
S	specific entropy, $\text{J/kg}^\circ\text{C}$
t	temperature, $^\circ\text{C}$
T	temperature, K
U	specific internal energy, J/kg
v	specific volume ($1/\rho$), m^3/kg
α	thermal expansion coefficient $-(\partial\rho/\partial t)_p/\rho$, $^\circ\text{C}^{-1}$
β	compressibility $(\partial\rho/\partial p)_t/\rho$, Pa^{-1}
ψ	specific Helmholtz free energy, J/kg
ρ	density, kg/m^3
μ	dynamic viscosity, Pa s

subscripts:

as	amorphous silica
chl	chalcedony
q	quartz
s	steam
w	liquid water
ac	α -cristobalite
β c	β -cristobalite

Table 2: Arguments for Option Parameter NOP in Main Program H20

NOP	For the Known Parameters	Supply Initial Guesses of
1	temperature & density	
2	temperature & pressure	density
3	pressure & density	temperature
4	density & enthalpy	temperature
5	temperature & enthalpy	density
6	entropy & density	temperature
7	entropy & temperature	density
8	entropy & pressure	temperature & density
9	enthalpy & pressure	temperature & density
10	entropy & enthalpy	temperature & density

guesses of temperature and density until the calculated values of pressure, enthalpy and entropy match their input values to within a user-supplied error tolerance (ERR). The default value of ERR is set in H20DR to 10^{-5} , or 1 part in 10^5 . If temperature and density are entered as known parameters, then no root-solving is necessary. Within H20DR variables are converted to double precision, and these are passed upon calls to H201 and H202.

H201 (input: T, v; output: p) All parameters passed to and from H201 are double-precision floating-point numbers. Using the equation of state from Keenan et al. (1978), H201 solves for pressure from temperature and density. H201 also calculates and stores certain quantities that are used in H202 and therefore, must always be called before H202.

H202 (input: T, v, p; output: α , β , ψ , U, C_v , C_p , ...) All parameters passed to and from H202 are double-precision floating-point numbers. This subroutine completes the calculation of the important equation-of-state properties presented in Keenan et al. (1978) and the viscosity equations of Watson et al. (1980). All coefficients used in H201 and in the remaining thermodynamic and viscosity equations are included in block-data form at the end of H202.

SATH20 (MP) Whereas H20 presents properties of a single fluid phase, SATH20 does so for each phase in a coexisting water-steam system. Only a saturation temperature (NOS = 1) or a saturation pressure (NOS = 2) need be entered as input. SATH20 calls SATDR, SATROW and SATROS to obtain, respectively, saturation temperatures or pressures and initial estimates of water and steam densities which are then used by H2ODR to calculate the properties of each phase. Within ten degrees of the critical temperature, calculated properties may vary in the third significant digit from published values in Keenan et al. (1978); above 374°C, densities of both phases are set equal to the critical value. In addition to the properties listed as output for H20, SATH20 also prints out the heat, energy and entropy of vaporization, dp/dt along the saturation line, and the specific-volume ratio of steam to water.

SATDR (input: t or p, and NOS; output: p or t, and dp/dt) All parameters passed to and from SATDR are single-precision floating-point numbers, with the exception of NOS which is a integer number. Saturation pressure is expressed as a transcendental function of temperature (in SATTP), and so Reynolds (1979) provides this driver program to calculate temperature from a known pressure by a Newton-Raphson root-solving technique.

SATTP (input: T; output: p, dp/dT) All parameters passed to and from SATTP are single-precision floating-point numbers. The equation for the pressure of coexisting water and steam as a function of temperature is presented in Keenan et al. (1978) and is used in SATTP along with an expression for the derivative of this pressure with respect to temperature.

SATROW (input: t; output: ρ_w) All parameters passed to and from SATROW are single-precision floating-point numbers. SATROW uses an equation for the density of saturated liquid water as a function of temperature presented in Reynolds (1979). This value of density is used as an initial guess by subroutine H2ODR.

SATROS (input: p ; output: ρ_g) All parameters passed to and from SATROS are single-precision floating-point numbers. SATROS calculates the density of steam at a given saturation pressure. This value of density is used as an initial guess by the subroutine H2ODR.

SILSOL (MP) Either a temperature and pressure, or concentration and fluid density can be entered to calculate, respectively, the solubilities of the solid phases of SiO_2 , or the temperatures of saturation from the solubility equations. For either case, the fluid may be two-phase. SILSOL may call SILICA, TSIO2, SATDR, SATROW, SATROS or H2ODR, depending upon which program options are chosen.

SILICA (input: t , ρ , α , NS; output: k_- , s_q , $(\partial s_q / \partial t)_p$, s_{as} , $(\partial s_{as} / \partial t)_p$, s_{chl} , ds_{chl}/dt , s_{ac} , ds_{ac}/dt , $s_{\beta c}$, $ds_{\beta c}/dt$) All parameters passed to and from SILICA are single-precision floating-point numbers, with the exception of NS which is a integer number. Input parameters ρ and α are vectors of length two. For NS = 1, only a single-phase is present and the only the first elements of the vectors need be specified. For NS = 2, two phases are present, and the first values are for a liquid phase and the second for a steam phase. Output parameter k_- is a vector of length two, the first element being the rate constant in units of $\text{kg/m}^2\text{s}$ and the second being that in units of m/s . Upon output, solubilities and derivatives of solubilities for quartz and amorphous silica are passed in two parameters, QTZ and AMS, that are 2×8 matrices. For NS = 1 only the first columns are used; for NS = 2 water and steam are present, and both columns are used. The first four rows are solubilities in units of (1) moles of SiO_2 per kg of fluid, (2) moles of SiO_2 per liter of fluid, (3) mg of SiO_2 per kg of fluid, and (4) kg of H_4SiO_4 per m^3 of fluid, respectively; the last four rows are partial derivatives of solubility with respect to temperature at constant pressure in corresponding units. Solubilities and their derivatives for chalcedony, and α - and β -cristabolite are passed in three parameters, CHL, ACR and BCR, that are vectors of length 8. Solubility equations for these minerals are not applicable when steam is present. Equations in SILICA are from Rimstidt and Barnes (1980), Fournier (1981), Fournier and Potter (1982a), and Fournier (1984).

TSIO2 (input: c , ρ , NS; output: t_q , t_{as} , t_{chl} , t_{ac} , $t_{\beta c}$) All parameters passed to and from TSIO2 are single-precision floating-point numbers, with the exception of NS which is a integer number. In SILICA, solubilities are calculated from known values of temperature. In TSIO2, temperatures are calculated from known silica concentrations through inversion of the same solubility equations. Therefore, from one concentration value, five temperatures, representative of each silica phase, are calculated. For the quartz and amorphous silica "geothermometers", density of the fluid is a required input parameter, and temperature is solved by a quartic solution of the original version of the solubility equations presented in SILICA. If the system is two-phase, water density is not required, and initial guesses of temperature for quartz and amorphous silica are derived from solubility equations which are wholly temperature dependent. If concentration reflects that of the steam phase, its density must be included as an input parameter. Output temperatures are passed as a vector of length 5, corresponding to equilibrium temperatures for quartz, amorphous silica, chalcedony, and α - and β -cristabolite, respectively. Input concentrations for TSIO2 are in molal units.

User-Supplied FORTRAN Programs H2ODR is the only subroutine which needs to be called for the complete set of properties of water. SATDR is the only subroutine which needs to be called for the temperature and pressure of the two-phase curve. SATROW, and especially SATROS, do not give exact values for the densities of coexisting liquid and vapor. For a given temperature, density and thermal expansion coefficient, SILICA can be called for all solubilities, their derivatives with respect to temperature, and precipitation-rate constants. For a given molal silica concentration and density, TSIO2 can be called to calculate temperatures from all of the SiO_2 geothermometers. All subroutines, with the exceptions of H2O1 and H2O2, accept and return single-precision variables.

The programs are presently arranged to accept input from and display output to the FORTRAN file unit "*", which refers to the keyboard and CRT unit on many machines. (Many FORTRAN compilers use "5" and "6" for the keyboard and CRT or line printer.) The user can change to different input and output devices by rewriting the WRITE statements in these programs.

APPLICATION OF MAIN PROGRAMS

The examples below illustrate the use of the interactive programs. User-supplied entries are denoted by the symbol ">" at the beginning of a line.

>run h2o

enter 0 for nop, if you need to see the list of options

nop = ??

>0

```

nop = 1: enter temp & density
      = 2:      temp & press  ; guess density
      = 3:      press & density ;      temp
      = 4:      density & enthalpy;      temp
      = 5:      temp & enthalpy;      density
      = 6:      entropy & density ;      temp
      = 7:      entropy & temp  ;      density
      = 8:      entropy & press ;      temp & density
      = 9:      enthalpy & press ;      temp & density
      = 10:     entropy & enthalpy;      temp & density

```

nop = ??

>1

temp_C, rho = ?

>400, 600

```

0.40000E+03 = temperature_C
0.56085E+08 = pressure
0.60000E+03 = density
0.40953E-02 = thermal expansion coefficient
0.54903E-08 = compressibility
0.71127E-04 = dynamic viscosity
0.28269E+04 = heat capacity (constant volume)
0.62540E+04 = heat capacity (constant pressure)
0.18539E+07 = enthalpy
0.17604E+07 = internal energy
0.39570E+04 = entropy
-0.90322E+06 = Helmholtz free energy
-0.80974E+06 = Gibbs free energy

```

another ? (y/n)

>y

nop = ??

>2

temp_C, press_MKS; Guess density_ = ?

>400, 560.85E05, 700

0.40000E+03 = temperature_C
 0.56085E+08 = pressure
 0.60000E+03 = density
 0.40953E-02 = thermal expansion coefficient
 0.54903E-08 = compressibility
 0.71127E-04 = dynamic viscosity
 0.28269E+04 = heat capacity (constant volume)
 0.62540E+04 = heat capacity (constant pressure)
 0.18539E+07 = enthalpy
 0.17604E+07 = internal energy
 0.39570E+04 = entropy
 -0.90321E+06 = Helmholtz free energy
 -0.80974E+06 = Gibbs free energy

another ? (y/n)

>y

nop = ??

>9

enthalpy_MKS, press_; Guess temp_C, density_ = ?

>1.8539E6, 560.85e5, 500, 700

0.40000E+03 = temperature_C
 0.56085E+08 = pressure
 0.60000E+03 = density
 0.40953E-02 = thermal expansion coefficient
 0.54902E-08 = compressibility
 0.71127E-04 = dynamic viscosity
 0.28269E+04 = heat capacity (constant volume)
 0.62540E+04 = heat capacity (constant pressure)
 0.18539E+07 = enthalpy
 0.17604E+07 = internal energy
 0.39570E+04 = entropy
 -0.90321E+06 = Helmholtz free energy
 -0.80974E+06 = Gibbs free energy

another ? (y/n)

>n

run sath2o

nos = 1 for given temp
 = 2 for given press

>1

temp_C = ??

>200

water	steam	
0.20000E+03	0.20000E+03	= temperature_C
0.15538E+07	0.15538E+07	= pressure
0.86469E+03	0.78519E+01	= density
0.13762E-02	0.32301E-02	= thermal expansion coefficient
0.90088E-09	0.72206E-06	= compressibility
0.13503E-03	0.15712E-04	= dynamic viscosity
0.33428E+04	0.19236E+04	= heat capacity (constant volume)
0.44931E+04	0.27944E+04	= heat capacity (constant pressure)
0.85245E+06	0.27932E+07	= enthalpy
0.85065E+06	0.25950E+07	= internal energy
0.23309E+04	0.64323E+04	= entropy
-0.25221E+06	-0.44812E+06	= Helmholtz free energy
-0.25041E+06	-0.25023E+06	= Gibbs free energy

0.19407E+07 = heat of vaporization
 0.17447E+07 = energy of vaporization
 0.41014E+04 = entropy of vaporization
 0.32494E+05 = (dp/dt)sat
 0.11013E+03 = V_steam/V_water

another ?? (y/n)

>y

nos = 1 for given temp
 = 2 for given press and trial temp

>2

press_MKS = ??

>15.538E05

water	steam	
0.20000E+03	0.20000E+03	= temperature_C
0.15538E+07	0.15538E+07	= pressure
0.86469E+03	0.78519E+01	= density
0.13762E-02	0.32301E-02	= thermal expansion coefficient
0.90088E-09	0.72206E-06	= compressibility
0.13503E-03	0.15712E-04	= dynamic viscosity
0.33428E+04	0.19236E+04	= heat capacity (constant volume)
0.44931E+04	0.27944E+04	= heat capacity (constant pressure)
0.85245E+06	0.27932E+07	= enthalpy
0.85065E+06	0.25953E+07	= internal energy
0.23309E+04	0.64323E+04	= entropy
-0.25221E+06	-0.44812E+06	= Helmholtz free energy
-0.25041E+06	-0.25023E+06	= Gibbs free energy

0.19407E+07 = heat of vaporization
 0.17447E+07 = energy of vaporization
 0.41014E+04 = entropy of vaporization
 0.32494E+05 = (dp/dt)sat
 0.11013E+03 = V_steam/V_water

another ?? (y/n)

>n

>run silsol

Do you want to enter temperature or concentration? (t/c)

>t

Two-phase fluid (water and steam)? (y/n)

>n

temp_C, press_Pa, rho(guess)_kg/m**3 = ??

>200, 100E05, 900

temp = 0.2000E+03 C
press = 0.1000E+08 Pa

Precipitation-rate constant
k_ = 0.6341E-06 kg/m*m s k_ = 0.7279E-09 m/s

	Molal mol(SiO2)/kg(f)	Molar mol(SiO2)/l(f)	ppm mg(SiO2)/kg(f)	H4SiO4 density kg(H4SiO4)/m**3(f)
Solubility quartz	0.4461E-02	0.3886E+01	0.2681E+03	0.3735E+00
am silica	0.1556E-01	0.1356E+02	0.9352E+03	0.1303E+01
chalcedony	0.5372E-02	0.4679E+01	0.3228E+03	0.4498E+00
a-cristob	0.7722E-02	0.6727E+01	0.4640E+03	0.6466E+00
b-cristob	0.1204E-01	0.1049E+02	0.7234E+03	0.1008E+01

(d(Sol))/dT)p				
quartz	0.5827E-04	0.4565E-01	0.3501E+01	0.4387E-02
am silica	0.1126E-03	0.8028E-01	0.6767E+01	0.7716E-02

d(Sol)/dT				
chalcedony	0.5703E-04	0.4968E-01	0.3426E+01	0.4775E-02
a-cristob	0.7944E-04	0.6920E-01	0.4773E+01	0.6651E-02
b-cristob	0.9673E-04	0.8426E-01	0.5812E+01	0.8099E-02

another? (y/n)

>y

Do you want to enter temperature or concentration? (t/c)

>c

Two-phase fluid (water and steam)? (y/n)

>n

Molal, molar, or ppm? (1/2/3)

>1

Silica concentration, density = ??

>0.4461E-02, 871.07

	Molal mol(SiO ₂)/kg(f)	Molar mol(SiO ₂)/l(f)	ppm mg(SiO ₂)/kg(f)	H ₄ SiO ₄ density kg(H ₄ SiO ₄)/m**3(f)
concentration	0.4461E-02	0.3886E+01	0.2680E+03	0.3735E+00

	temperature_C	pressure_Pa
quartz	0.2000E+03	0.9995E+07
amorphous silica	0.7631E+02	-0.1105E+09

	temperature_C
chalcedony	0.1831E+03
a-cristobalite	0.1521E+03
b-cristobalite	0.1020E+03

another? (y/n)

>y

Do you want to enter temperature or concentration? (t/c)

>t

Two-phase fluid (water and steam)? (y/n)

>y

temp_C of coexisting water and steam = ??

>200

NOTE: Values are for liquid phase only

temp = 0.2000E+03 C

press = 0.1554E+07 Pa

Precipitation-rate constant

k_ = 0.6341E-06 kg/m*m s k_ = 0.7333E-09 m/s

Solubility	Molal mol(SiO ₂)/kg(f)	Molar mol(SiO ₂)/l(f)	ppm mg(SiO ₂)/kg(f)	H ₄ SiO ₄ density kg(H ₄ SiO ₄)/m**3(f)
quartz	0.4385E-02	0.3792E+01	0.2635E+03	0.3644E+00
am silica	0.1550E-01	0.1340E+02	0.9312E+03	0.1288E+01
chalcedony	0.5372E-02	0.4645E+01	0.3228E+03	0.4465E+00
a-cristob	0.7722E-02	0.6677E+01	0.4640E+03	0.6418E+00
b-cristob	0.1204E-01	0.1041E+02	0.7234E+03	0.1001E+01
(d(Sol)/dT) _p				
quartz	0.5671E-04	0.4382E-01	0.3408E+01	0.4212E-02
am silica	0.1109E-03	0.7749E-01	0.6666E+01	0.7448E-02
d(Sol)/dT				
chalcedony	0.5703E-04	0.4931E-01	0.3426E+01	0.4740E-02
a-cristob	0.7944E-04	0.6869E-01	0.4773E+01	0.6603E-02
b-cristob	0.9673E-04	0.8364E-01	0.5812E+01	0.8040E-02

another? (y/n)

>y

Do you want to enter temperature or concentration? (t/c)

>c

Two-phase fluid (water and steam)? (y/n)

>y

Is concentration that of steam phase? (y/n)

>n

Molal or ppm? (1/2)

>2

Silica concentration (liquid water) = ??

>322.8

	Molal mol(SiO ₂)/kg(f)	Molar mol(SiO ₂)/l(f)	ppm mg(SiO ₂)/kg(f)	H ₄ SiO ₄ density kg(H ₄ SiO ₄)/m**3(f)
concentration	0.5372E-02		0.3228E+03	

	temperature_C	pressure_Pa
quartz	0.2164E+03	0.2161E+07
amorphous silica	0.9034E+02	0.7104E+05

	temperature_C
chalcedony	0.2000E+03
a-cristobalite	0.1672E+03
b-cristobalite	0.1171E+03

another? (y/n)

>y

Do you want to enter temperature or concentration? (t/c)

>t

Two-phase fluid (water and steam)? (y/n)

>y

temp_C of coexisting water and steam = ??

>300

Note: Values are for liquid phase, unless noted

temp = 0.3000E+03 C
press = 0.8581E+07 Pa

Precipitation-rate constant
k_ = 0.5756E-05 kg/m*m s k_ = 0.8080E-08 m/s

	Molal mol(SiO ₂)/kg(f)	Molar mol(SiO ₂)/l(f)	ppm mg(SiO ₂)/kg(f)	H ₄ SiO ₄ density kg(H ₄ SiO ₄)/m**3(f)
Solubility				
qtz: water	0.1130E-01	0.8054E+01	0.6792E+03	0.7741E+00
steam	0.6491E-04	0.2995E-02	0.3900E+01	0.2878E-03
ams: water	0.2604E-01	0.1855E+02	0.1565E+04	0.1783E+01
steam	0.2012E-02	0.9281E-01	0.1209E+03	0.8920E-02
chalcedony	n/a	n/a	n/a	n/a
a-cristob	n/a	n/a	n/a	n/a
b-cristob	n/a	n/a	n/a	n/a
(d(Sol)/dT) _p				
qtz: water	0.5415E-04	0.1204E-01	0.3253E+01	0.1157E-02
steam	0.3185E-06	-0.6473E-05	0.1914E-01	-0.6222E-06
ams: water	0.7295E-04	-0.9158E-02	0.4383E+01	-0.8802E-03
steam	-0.1058E-04	-0.1144E-02	-0.6355E+00	-0.1100E-03
d(Sol)/dT				
chalcedony	n/a	n/a	n/a	n/a
a-cristob	n/a	n/a	n/a	n/a
b-cristob	n/a	n/a	n/a	n/a

another? (y/n)

>y

Do you want to enter temperature or concentration? (t/c)

>c

Two-phase fluid (water and steam)? (y/n)

>y

Is concentration that of steam phase? (y/n)

>y

Molal, molar, or ppm? (1/2/3)

>2

Silica concentration, density = ??

>0.9281E-01, 46.137

	Molal mol(SiO ₂)/kg(f)	Molar mol(SiO ₂)/l(f)	ppm mg(SiO ₂)/kg(f)	H ₄ SiO ₄ density kg(H ₄ SiO ₄)/m**3(f)
concentration	0.2012E-02	0.9281E-01	0.1209E+03	0.8920E-02

	temperature_C	pressure_Pa
quartz	n/a	n/a
amorphous silica	0.3000E+03	0.8581E+07

another? (y/n)

>n

REFERENCES

- Fournier, R.O., 1981, Application of water geochemistry to geothermal exploration and reservoir engineering: Chapter 4 in Geothermal Systems: Principles and Case Histories, L. Ryback and L.J.P. Muffler, eds., John Wiley, New York, pp. 109-143.
- Fournier, R.O., 1984, personal communication.
- Fournier, R.O., and W.L. Marshall, 1983, Calculation of amorphous silica solubilities at 25° to 300°C and apparent cation hydration numbers in aqueous salt solutions using the concept of effective density of water: *Geochimica et Cosmochimica Acta*, v. 47, pp. 587-596.
- Fournier, R.O., and R.W. Potter, II, 1982a, An equation correlating the solubility of quartz in water from 25° to 900°C at pressures up to 10,000 bars: *Geochimica et Cosmochimica Acta*, v. 46, pp. 1969-1973.
- Fournier, R.O., and R.W. Potter, II, 1982b, A revised and expanded silica (quartz) geothermometer: *Geothermal Resources Council Bulletin*, v. 11, pp. 3-12.
- Keenan, J.H., F.G. Keyes, P.G. Hill and J.G. Moore, 1978, *Steam Tables: Thermodynamic Properties of Water Including Vapor, Liquid, and Solid Phases*, John Wiley, New York, 156 p.
- Meissner, L.P., and E.I. Organick, 1980, *Fortran 77: Featuring Structured Programming*, Addison-Wesley, Reading, Mass., 500 p.
- Reynolds, W.C., 1979, *Thermodynamic Properties in SI*, Department of Mechanical Engineering, Stanford University, 173 p.
- Rimstidt, T.D., and H.L. Barnes, 1980, The kinetics of silica-water reactions: *Geochimica et Cosmochimica Acta*, v. 44, pp. 1683-1699.
- Watson, J.T.R., R.S. Basu and J.V. Sengers, 1980, An improved equation for the viscosity of water and steam, in *Water and Steam: Their Properties and Current Industrial Applications*, J. Straub and K. Scheffler, eds., Pergamon Press, New York, pp. 336-343.

PROGRAM LISTINGS

Main Program H2O

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

PROGRAM H2O

C

C THERMODYNAMIC PROPERTIES OF WATER FROM KEENAN ET AL (1978)

C

C INPUT IS ANY TWO AMONG TEMP, PRESS, DENSITY, ENTROPY, ENTHALPY

C

C EXTERNAL SUBROUTINES: H2ODR, H2O1, H2O2

C

IMPLICIT REAL*4 (A-H,O-Z)

EXTERNAL H2ODR

CHARACTER*1 ANS

DATA ERR/1.0E-05/

C

C READ INPUT PARAMETERS

C

WRITE (*,105)

4 WRITE (*,107)

READ (*,*) NOP

IF (NOP .NE. 0) GOTO 3

WRITE (*,110)

GOTO 4

3 GOTO (11,12,13,14,15,16,17,18,19,20), NOP

11 WRITE (*,111)

READ (*,*) T, RO

GOTO 1

12 WRITE (*,112)

READ (*,*) T, P, RO

GOTO 1

13 WRITE (*,113)

READ (*,*) P, RO, T

GOTO 1

14 WRITE (*,114)

READ (*,*) RO, H, T

GOTO 1

15 WRITE (*,115)

READ (*,*) T, H, RO

GOTO 1

16 WRITE (*,116)

READ (*,*) S, RO, T

GOTO 1

17 WRITE (*,117)

READ (*,*) S, T, RO

GOTO 1

18 WRITE (*,118)

READ (*,*) S, P, T, RO

GOTO 1

19 WRITE (*,119)

READ (*,*) H, P, T, RO

```

      GOTO 1
20   WRITE (*,120)
      READ (*,*) S, H, T, RO
C
C   CALCULATE PROPERTIES OF H2O
C
1    CALL H2ODR(T,P,RO,ALFA,BETA,PSI,U,CV,CP,H,G,S,VISC,NOP,ERR)
C
C   WRITE OUTPUT
C
      WRITE (*,130) T,P,RO,ALFA,BETA,VISC,CV,CP,H,U,S,PSI,G
C
C   PROMPT USER FOR ANOTHER VALUE OR EXIT
C
      WRITE (*,140)
      READ (*,150) ANS
      IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) GOTO 4
C
C   FORMATS
C
105  FORMAT (/1X,'enter 0 for nop,',
.      ' if you need to see the list of options'//)
107  FORMAT (5X,'nop = ??'//)
110  FORMAT (/1X,'nop = 1: enter temp & density'
.      /1X,'      = 2:      temp & press   ; guess density'
.      /1X,'      = 3:      press & density ;      temp'
.      /1X,'      = 4:      density & enthalpy;      temp'
.      /1X,'      = 5:      temp & enthalpy;      density'
.      /1X,'      = 6:      entropy & density ;      temp'
.      /1X,'      = 7:      entropy & temp    ;      density'
.      /1X,'      = 8:      entropy & press   ;      temp & density'
.      /1X,'      = 9:      enthalpy & press  ;      temp & density'
.      /1X,'      = 10:     entropy & enthalpy;      temp & density'//)
111  FORMAT (5X,'temp_C, density = ?'//)
112  FORMAT (5X,'temp_C, press_MKS; Guess density_ = ?'//)
113  FORMAT (5X,'press_MKS, density_; Guess temp_C = ?'//)
114  FORMAT (5X,'density_MKS, enthalpy_; Guess temp_C = ?'//)
115  FORMAT (5X,'temp_C, enthalpy_MKS; Guess density_ = ?'//)
116  FORMAT (5X,'entropy_MKS, density; Guess temp_C = ?'//)
117  FORMAT (5X,'entropy_MKS, temp_C; Guess density_ = ?'//)
118  FORMAT (5X,'entropy_MKS, press_; Guess temp_C, density_ = ?'//)
119  FORMAT (5X,'enthalpy_MKS, press_; Guess temp_C, density_ = ?'//)
120  FORMAT (5X,'entropy_MKS, enthalpy_; Guess temp_C, density_ = ?'//)
130  FORMAT (/5X,E15.5,' = temperature_C'
.      /5X,E15.5,' = pressure'
.      /5X,E15.5,' = density'
.      /5X,E15.5,' = thermal expansion coefficient'
.      /5X,E15.5,' = compressibility'
.      /5X,E15.5,' = dynamic viscosity'
.      /5X,E15.5,' = heat capacity (constant volume)'
.      /5X,E15.5,' = heat capacity (constant pressure)'
.      /5X,E15.5,' = enthalpy'
.      /5X,E15.5,' = internal energy'
.      /5X,E15.5,' = entropy'

```

```
      .      /5X,E15.5,' = Helmholtz free energy'  
      .      /5X,E15.5,' = Gibbs free energy'//  
140  FORMAT (1X,'another ? (y/n)'/)  
150  FORMAT (A1)  
      STOP  
      END
```

Main Program SATH20

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2
C

PROGRAM SATH20

C
C PROPERTIES OF SATURATED WATER, GIVEN TEMP OR PRESS
C

C EQ. OF STATE FROM KEENAN ET AL (1978)
C

C EXTERNAL SUBROUTINES: SATDR, SATTP, SATROS, SATROW,
C H2ODR, H2O1, H2O2
C

IMPLICIT REAL*4 (A-H,O-Z)

EXTERNAL SATDR, SATROW, SATROS, H2ODR

CHARACTER*1 ANS

DATA TC, PC, ROC /647.286E0, 220.8935869E5, 317.0E0/

C
C READ INPUT PARAMETERS AND BEGIN BY INITIALLIZING
C

4 TS = 0.0E0
PS = 0.0E0
NOP = 2
ERR = 1.0E-05
WRITE (*,100)
READ (*,*) NOS
GOTO (1,2), NOS
1 WRITE (*,102)
READ (*,*) TS
GOTO 3
2 WRITE (*,104)
READ (*,*) PS

C
C CHECK IF TEMPERATURE OR PRESSURE IS GREATER THAN CRITICAL POINT
C

3 IF ((TS .GT. TC-273.15E0) .OR. (PS .GT. PC)) GOTO 6
C

C CALCULATE PRESSURE OR TEMPERATURE BY CALLING DRIVER PROGRAM
C

CALL SATDR(TS,PS,DPDTS,NOS)

C
C SET DENSITIES NEAR CRITICAL POINT TO CRITICAL VALUE
C

IF (TS .LT. 374.001E0) GOTO 7
ROL = ROC
ROV = ROC
NOP = 1
GOTO 8

C
C AWAY FROM CRITICAL POINT, CALCULATE DENSITIES OF EACH PHASE
C

7 CALL SATROW(TS,ROL)
CALL SATROS(PS,ROV)

C

```

C  DECREASE ERROR PARAMETER FOR H2ODR WHEN APPROACHING CRIT. POINT
C
8  IF (TS .GE. 360.OE0) ERR = 1.OE-07
    CALL H2ODR
    . (TS,PS,ROL,ALFAL,BETAL,PSIL,UL,CVL,CPL,HL,GSL,SL,XMUL,NOP,ERR)
    CALL H2ODR
    . (TS,PS,ROV,ALFAV,BETA,PSIV,UV,CVV,CPV,HV,GSV,SV,XMUV,NOP,ERR)
C
C  CALCULATE ENERGIES & ENTROPIES OF VAPORIZATION, AND VOLUME RATIO
C
    SLV = SV - SL
    HLV = HV - HL
    ULV = UV - UL
    VOL = ROL/ROV
C
C  WRITE OUTPUT
C
    WRITE (*,120)
    WRITE (*,130) TS,TS,PS,PS,ROL,ROV,ALFAL,ALFAV,BETAL,BETA,
    . XMUL,XMUV,CVL,CVV,CPL,CPV,HL,HV,UL,UV,SL,SV,PSIL,PSIV,GSL,GSV
    WRITE (*,140) HLV, ULV, SLV, DPDT, VOL
    GOTO 5
6  WRITE (*,160)
C
C  PROMPT USER FOR ANOTHER VALUE, OR EXIT
C
5  WRITE (*,145)
    READ (*,155) ANS
    IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) GOTO 4
C
C  FORMATS
C
100  FORMAT (5X,'nos = 1 for given temp'/
    . 5X,' = 2 for given press'/)
102  FORMAT (5X,'temp_C = ??'/)
104  FORMAT (5X,'press_MKS = ??'/)
120  FORMAT (/12X,'water',11X,'steam')
130  FORMAT (/5X,E15.5,E15.5,' = temperature_C'
    . /5X,E15.5,E15.5,' = pressure'
    . /5X,E15.5,E15.5,' = density'
    . /5X,E15.5,E15.5,' = thermal expansion coefficient'
    . /5X,E15.5,E15.5,' = compressibility'
    . /5X,E15.5,E15.5,' = dynamic viscosity'
    . /5X,E15.5,E15.5,' = heat capacity (constant volume)'
    . /5X,E15.5,E15.5,' = heat capacity (constant pressure)'
    . /5X,E15.5,E15.5,' = enthalpy'
    . /5X,E15.5,E15.5,' = internal energy'
    . /5X,E15.5,E15.5,' = entropy'
    . /5X,E15.5,E15.5,' = Helmholtz free energy'
    . /5X,E15.5,E15.5,' = Gibbs free energy')
140  FORMAT (/1X,E15.5,' = heat of vaporization'
    . /1X,E15.5,' = energy of vaporization'
    . /1X,E15.5,' = entropy of vaporization'
    . /1X,E15.5,' = (dp/dt)sat'

```

```
      /1X,E15.5,' = V_steam/V_water')  
145  FORMAT (/1X,'another ??'(y/n) '/')  
155  FORMAT (A1)  
160  FORMAT (/1X,'temperature or pressure exceeds critical value'/)  
      STOP  
      END
```

Main Program SILSOL

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

PROGRAM SILSOL

C

C SOLUBILITIES AND KINETIC RATE CONSTANT FOR SILICA PHASES

C FROM FOURNIER & POTTER (1982),

C FOURNIER (1981,1984), RIMSTIDT & BARNES (1980)

C

C EXTERNAL SUBROUTINES: SILICA, TSIO2, H2ODR, H2O1, H2O2,

C SATDR, SATROW, SATROS, SATTP

C

IMPLICIT REAL*4 (A-H,L,O-Z)

CHARACTER*1 TSANS, ANS

DIMENSION RK(2), QTZ(2,8), AMS(2,8), CHL(8), ACR(8), BCR(8),

. RHO(2), NA(5), TS(5), PS(5), SOL(4), ALFA(2)

EXTERNAL H2ODR, SILICA, TSIO2, SATDR, SATROW, SATROS

DATA AWS,RHO(1),AWH,TC /6.0084E4,1.0E3,9.6115E-2,374.136E0/

C

C INITIALIZE PARAMETERS FOR SUCCESSIVE RUNS

C

1 DO 2 J = 1, 5

2 NA(J) = 0

C

C PROMPT USER AS TO WHICH CASE IS DESIRED:

C CASE 1: ENTER TEMPERATURE (TSANS = 'T')

C CASE 2: ENTER CONCENTRATION (TSANS = 'C')

C

WRITE (*,100)

100 FORMAT (/1X,'Do you want to enter temperature or
'concentration? (t/c)'/)

READ (*,110) TSANS

110 FORMAT (A1)

C

C PROMPT USER AS TO WHICH CASE IS DESIRED:

C CASE A: SYSTEM IS 1-PHASE (NS = 1)

C CASE B: SYSTEM IS 2-PHASE (NS = 2)

C

WRITE (*,120)

120 FORMAT (/1X,'Two-phase fluid ',
'(coexisting water and steam)? (y/n)'/)

READ (*,110) ANS

IF ((ANS .EQ. 'N') .OR. (ANS .EQ. 'n')) NS = 1

IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'y')) NS = 2

C

C IF CASE 1A (TEMP INPUT, 1-PHASE): TSANS = T, NS = 1, GO DOWN TO 21

C IF CASE 1B (TEMP INPUT, 2-PHASE): TSANS = T, NS = 2, GO DOWN TO 20

C

IF ((TSANS .EQ. 'T') .OR. (TSANS .EQ. 't')) GOTO (21, 20), NS

C

C IF CASE 2B (CONC INPUT, 2-PHASE): TSANS = C, NS = 2,

C PROMPT USER TO ACERTAIN IF DATA IS FOR WATER OR STEAM

C PROMPT USER TO GET UNITS OF CONCENTRATION

```

C      MOLAL OR PPM IF DATA IS FOR THE LIQUID PHASE
C      MOLAL, PPM, OR MOLAR IF DATA IS FOR STEAM PHASE
C
      IF (NS .EQ. 2) THEN
        WRITE(*,150)
150      FORMAT(/1X,'Is concentration that of steam phase? (y/n) '/')
        READ(*,110) ANS
        IF ((ANS .EQ. 'y') .OR. (ANS .EQ. 'Y')) THEN
          NS = 3
          WRITE (*,180)
180      FORMAT (/1X,'Molal, molar, or ppm? (1/2/3) '/')
          READ (*,*) NAS
          GOTO 25
        ELSE
160      WRITE (*,160)
          FORMAT (/1X,'Molal or ppm? (1/2) '/')
          READ (*,*) NAS
          IF (NAS .EQ. 2) NAS = 3
          GOTO 24
        ENDIF
C
C      IF CASE 2A (CONC INPUT, 1-PHASE): TSANS = C, NS = 1
C      PROMPT USER TO GET UNITS OF CONCENTRATION: MOLAL, MOLAR OR PPM
C
      ELSE
        WRITE (*,180)
        READ (*,*) NAS
        GOTO 25
      ENDIF
C
C      END OF SELECTION OF INPUT OPTIONS
C


---


C      CASE 1 (TEMPERATURE INPUT) ALWAYS RESULTS IN A CALL TO THE
C      SUBROUTINE "SILICA"
C
C      CASE 1B (TEMPERATURE INPUT, 2-PHASE): GET INPUT TEMPERATURE
C
20      WRITE (*,130)
130      FORMAT (/1X,'temp_C of coexisting water and steam = ??'/)
        READ (*,*) T
        IF (T .GT. 374.136E0) GOTO 18
C
C      FOR CASE 1B AND TEMPERATURES LESS THAN 250 C,
C      SOLUBILITY EQUATIONS ARE NOT VALID FOR STEAM PHASE
C      WARN USERS OF THIS, AND REVERT TO CASE 1A BECAUSE DATA FOR
C      ONLY ONE PHASE WILL BE CALCULATED
C
      IF (T .LT. 250.0E0) NS = 1
      IF (T .LT. 250.0E0) THEN
        WRITE (*,510)
      ELSE
        WRITE (*,520)
      ENDIF

```

510 FORMAT (/IX,'NOTE: Values are for liquid phase only'/)
 520 FORMAT (/IX,'NOTE: Values are for liquid phase, unless noted'/)

C
 C
 C

CASE 1B: PRESSURES AND DENSITIES OF BOTH FLUID PHASES ARE CALCULATED

CALL SATDR(T,P,DPDT,NP)
 CALL SATROW(T,RHO(1))
 IF (T .LT. 250.0E0) GOTO 23
 CALL H2ODR(T,P,RHO(1),ALFA(1),BETA,PSI,U,CV,CP,H,G,S,VIS,2,1.E-5)
 CALL SATROS(P,RHO(2))
 CALL H2ODR(T,P,RHO(2),ALFA(2),BETA,PSI,U,CV,CP,H,G,S,VIS,2,1.E-5)
 CALL SILICA(T,RHO,ALFA,RK,QTZ,AMS,CHL,ACR,BCR,NS)

C
 C

CASE 1B: CHECK APPLICABLE RANGES OF GIVEN TEMPERATURE & PRESSURE

NA(1) IS FOR RATE CONSTANT, WHICH MUST BE < 300 DEG C
 < 50 MPA (500 BARS)
 NA(2) IS FOR SOLUBILITY OF SILICA, MUST BE < 340 DEG C
 < 100 MPA

BECAUSE CASE 1B CAN ONLY BE REACHED IF T > 250, AND THE EQUATIONS FOR
 SOLUBILITIES OF CHALCEDONY AND ALFA- AND BETA-CRISTABOLITE
 ONLY APPLY IF < 250 DEG C, THESE DATA ARE NOT PRINTED OUT.

C

IF ((T .GT. 300.01E0) .OR. (P. GT. 5.0E7)) NA(1) = 1
 IF ((T .GT. 340.01E0) .OR. (P. GT. 1.0E8)) NA(2) = 1

C

WRITE OUTPUT FOR CASE 1B — TEMP, PRESSURE, RATE CONSTANTS
 AND HEADING FOR SOLUBILITY TABLE

C

WRITE (*,200) T
 IF (NA(1) .EQ. 1) THEN
 WRITE (*,210) P
 ELSE
 WRITE(*,220) P, RK(1), RK(2)
 ENDIF
 WRITE (*,230)

C

CASE 1B (TEMPERATURE INPUT, 2-PHASE): WRITE SOLUBILITY OUTPUT
 (NOTE THAT THIS OUTPUT CAN ONLY BE READ IF TEMPERATURE > 250 C)
 "N/A" WRITTEN IF TEMPERATURES ARE NOT APPLICABLE.

C

WRITE (*,330) (QTZ(J,1),QTZ(J,2),QTZ(J,3),QTZ(J,4), J=1,2)
 IF (NA(2) .NE. 1) THEN
 WRITE (*,340) (AMS(J,1),AMS(J,2),AMS(J,3),AMS(J,4), J=1,2)
 ELSE
 WRITE (*,260)
 ENDIF
 WRITE (*,300)
 WRITE (*,310)
 WRITE (*,330) (QTZ(J,5),QTZ(J,6),QTZ(J,7),QTZ(J,8), J=1,2)
 IF (NA(2) .NE. 1) THEN
 WRITE (*,340) (AMS(J,5),AMS(J,6),AMS(J,7),AMS(J,8), J=1,2)
 ELSE
 WRITE (*,260)

```

      ENDIF
      WRITE (*,350)
      WRITE (*,300)
      GOTO 19

C
C CASE 1A (TEMPERATURE INPUT, 1-PHASE):
C GET INPUT TEMPERATURE AND PRESSURE, & GUESS FLUID DENSITY
C
21  WRITE (*,140)
140  FORMAT (/1X,'temp_C, press_Pa, rho(guess)_kg/m**3 = ??'/)
      READ (*,*) T, P, RHO(1)
      IF ((T .GT. 900.0E0) .OR. (P. GT. 1.0E8)) GOTO 18

C
C CASE 1A: DENSITY IS CALCULATED
C
23  CALL H2ODR(T,P,RHO(1),ALFA(1),BETA,PSI,U,CV,CP,H,G,S,VIS,2,1.E-5)
      CALL SILICA(T,RHO,ALFA,RK,QTZ,AMS,CHL,ACR,BCR,NS)

C
C CASE 1A: CHECK APPLICABLE RANGES OF GIVEN TEMPERATURE & PRESSURE
C
C NA(1) IS FOR RATE CONSTANT, WHICH MUST BE < 300 DEG C
C                                     < 50 MPA (500 BARS)
C NA(2) IS FOR SOLUBILITY OF SILICA, MUST BE < 340 DEG C
C                                     < 100 MPA
C NA(3) IS FOR SOLUBILITY OF CHALCEDONY AND
C ALFA- AND BETA-CRISTABOLITE < 250 DEG C
C
      IF ((T .GT. 300.01E0) .OR. (P. GT. 5.0E07)) NA(1) = 1
      IF ((T .GT. 340.01E0) .OR. (P. GT. 1.0E8)) NA(2) = 1
      IF (T .GT. 250.01E0) NA(3) = 1

C
C WRITE OUTPUT FOR CASE 1A — TEMP, PRESSURE, RATE CONSTANTS
C AND HEADING FOR SOLUBILITY TABLE
C
      WRITE (*,200) T
      IF (NA(1) .EQ. 1) THEN
        WRITE (*,210) P
      ELSE
        WRITE(*,220) P, RK(1), RK(2)
      ENDIF
      WRITE (*,230)

C
C CASE 1A (TEMPERATURE INPUT, 1-PHASE): WRITE SOLUBILITY OUTPUT
C "N/A" WRITTEN IF TEMPERATURES ARE NOT APPLICABLE.
C
      WRITE (*,240) (QTZ(1,J), J=1,4)
      IF (NA(2) .NE. 1) THEN
        WRITE (*,250) (AMS(1,J), J=1,4)
      ELSE
        WRITE (*,260)
      ENDIF
      IF (NA(3) .NE. 1) THEN
        WRITE (*,270) (CHL(J), J=1,4)
        WRITE (*,280) (ACR(J), J=1,4)
      ENDIF

```

```

      WRITE (*,290) (BCR(J), J=1,4)
    ELSE
      WRITE (*,300)
    ENDIF
    WRITE (*,310)
    WRITE (*,240) (QTZ(1,J), J=5,8)
    IF (NA(2) .NE. 1) THEN
      WRITE (*,250) (AMS(1,J), J=5,8)
    ELSE
      WRITE (*,260)
    ENDIF
    WRITE (*,320)
    WRITE (*,350)
    IF (NA(3) .NE. 1) THEN
      WRITE (*,270) (CHL(J), J=5,8)
      WRITE (*,280) (ACR(J), J=5,8)
      WRITE (*,290) (BCR(J), J=5,8)
    ELSE
      WRITE (*,300)
    ENDIF
    GOTO 19

```

C

C FORMATS FOR DISPLAY OUTPUT OF CASE 1

C

```

330  FORMAT (3X,'qtz: water',2X,E11.4,3(5X,E11.4)/
      3X,'      steam',2X,E11.4,3(5X,E11.4))
340  FORMAT (3X,'ams: water',2X,E11.4,3(5X,E11.4)/
      3X,'      steam',2X,E11.4,3(5X,E11.4))
200  FORMAT (/2X,'temp =' ,E11.4,' C',17X,
      'Precipitation-rate constant')
210  FORMAT (2X,'press =' ,E11.4,' Pa',23X,' k_ =      n/a '/')
220  FORMAT (2X,'press =' ,E11.4,' Pa',7X,' k_ =' ,E11.4,' kg/m*m s',
      3X,' k_ =' ,E11.4,' m/s'/)
230  FORMAT (19X,'Molal',11X,'Molar',12X,'ppm',8X'H4SiO4 density'/2X,
      'Solubility mol(SiO2)/kg(f) mol(SiO2)/l(f) mg(SiO2)/kg(f) '
      'kg(H4SiO4)/m**3(f)')
240  FORMAT (3X,'quartz' ,2X,E11.4,3(5X,E11.4))
250  FORMAT (3X,'am silica' ,2X,E11.4,3(5X,E11.4))
260  FORMAT (3X,'am silica' ,7X,'n/a',3(13X,'n/a'))
270  FORMAT (3X,'chalcedony',2X,E11.4,3(5X,E11.4))
280  FORMAT (3X,'a-cristob' ,2X,E11.4,3(5X,E11.4))
290  FORMAT (3X,'b-cristob' ,2X,E11.4,3(5X,E11.4))
300  FORMAT (3X,'chalcedony',7X,'n/a',3(13X,'n/a'))/
      3X,'a-cristob' ,7X,'n/a',3(13X,'n/a'))/
      3X,'b-cristob' ,7X,'n/a',3(13X,'n/a'))
310  FORMAT (/2X,'(d(Sol)/dT)p')
320  FORMAT (1X)
350  FORMAT (2X,'d(Sol)/dT')

```

C

C END OF CASE 1

C

C

C CASE 2 (CONCENTRATION INPUT) ALWAYS RESULTS IN A CALL TO THE
 C SUBROUTINE "TSIO2"

```

C
C CASE 2B (CONCENTRATION INPUT, 2-PHASE): GET INPUT CONCENTRATION
C PUT CONC INTO MOLAL (NAS = 1) FROM PPM (NAS = 3)
C
24  WRITE (*,170)
170  FORMAT (/1X,'Silica concentration (liquid phase) = ??'/)
      READ (*,*) SOL(NAS)
      IF (NAS .EQ. 3) SOL(1) = SOL(3)/AWS
C
C TO GET A FIRST GUESS OF TEMPERATURE, CALL TSIO2 USING ONLY
C THE TEMPERATURE DEPENDENT EQUATIONS
C
      CALL TSIO2(TS,RHO,SOL(1),2)
C
C TWO DENSITIES ARE CALCULATED FOR THE PREDICTED TEMPERATURES;
C ONE FOR THE QUARTZ TEMP, AND ANOTHER FOR THE AMORPHOUS SILICA
C
      CALL SATROW(TS(1),RHO(1))
      CALL SATROW(TS(2),RHO(2))
C
C THESE DENSITIES ARE USED TO OBTAIN BETTER TEMPERATURE
C ESTIMATES FROM THE MORE COMPLETE SOLUBILITY EQUATIONS
C
      CALL TSIO2(TS,RHO,SOL(1),4)
      CALL SATDR(TS(1),PS(1),DPDT,1)
      CALL SATDR(TS(2),PS(2),DPDT,1)
      GOTO 5
C
C CASE 2A (CONCENTRATION INPUT, 1-PHASE): GET INPUT CONCENTRATION
C PUT CONC INTO MOLAL (NAS = 1) FROM MOLAR (= 2) OR PPM (= 3)
C
25  WRITE (*,190)
190  FORMAT (/1X,'Silica concentration, density = ??'/)
      READ (*,*) SOL(NAS), RHO(1)
      IF (NAS .EQ. 3) SOL(1) = SOL(3)/AWS
      IF (NAS .EQ. 2) SOL(1) = SOL(2)/RHO(1)
      SOL(2) = SOL(1)*RHO(1)
      SOL(3) = SOL(1)*AWS
      SOL(4) = SOL(1)*AWH*RHO(1)
      CALL TSIO2(TS,RHO,SOL(1),NS)
C
C USE PREDICTED TEMPERATURES & DENSITY TO CALCULATE PRESSURES
C FOR QUARTZ AND AMORPHOUS SILICA
C
      CALL H2ODR(TS(1),PS(1),RHO(1),ALFA(1),BETA,
      .          PSI,U,CV,CP,H,G,S,VIS,1,1.0E-5)
      CALL H2ODR(TS(2),PS(2),RHO(1),ALFA(1),BETA,
      .          PSI,U,CV,CP,H,G,S,VIS,1,1.0E-5)
C
C CASE 2: CHECK APPLICABLE RANGES OF TEMPERATURES
C
C NA(1) IS FOR QUARTZ TEMP, WHICH MUST BE < 374.136 DEG C
C NA(2) IS FOR AM-SILICA TEMP,           < 340 DEG C
C NA(3) IS FOR CHALCEDONY TEMP,           < 250 DEG C

```

```

C   NA(4)          ALFA-CRISTABOLITE          < 250 DEG C
C   NA(5)          BETA-CRISTABOLITE          < 250 DEG C
C
C   FOR NS = 3 (COEXISTING WATER AND STEAM, CONCENTRATIONS GIVEN
C   FOR STEAM PHASE), THERE ARE SPECIAL CONSTRAINTS
C
5  IF ((TS(1) .LT. 0.0E0) .OR. (TS(1) .GT. 900.0E0)) NA(1) = 1
    IF ((TS(2) .LT. 0.0E0) .OR. (TS(2) .GT. 340.0E0)) NA(2) = 1
    IF ((TS(3) .LT. 0.0E0) .OR. (TS(3) .GT. 250.0E0)) NA(3) = 1
    IF ((TS(4) .LT. 0.0E0) .OR. (TS(4) .GT. 250.0E0)) NA(4) = 1
    IF ((TS(5) .LT. 0.0E0) .OR. (TS(5) .GT. 250.0E0)) NA(5) = 1
    IF (NS .EQ. 3) THEN
      IF (TS(1) .GT. 374.136E0) NA(1) = 1
      IF (TS(1) .LT. 250.0E0) NA(1) = 1
      IF (TS(2) .LT. 250.0E0) NA(2) = 1
      TS(3) = 1
      TS(4) = 1
      TS(5) = 1
    ENDIF
    IF (NA(1)+NA(2)+NA(3)+NA(4)+NA(5) .EQ. 5) GOTO 18
C
C   CASE 2 (ENTERED CONCENTRATIONS): WRITE OUTPUT
C   "N/A" WRITTEN IF TEMPERATURES ARE NOT APPLICABLE.
C
    IF (NS .EQ. 2) THEN
      WRITE (*,375) SOL(1), SOL(3)
    ELSE
      WRITE (*,370) (SOL(J), J=1,4)
    ENDIF
    IF (NA(1) .NE. 1) THEN
      WRITE (*,380) TS(1), PS(1)
    ELSE
      WRITE (*,500)
    ENDIF
    IF (NA(2) .NE. 1) THEN
      WRITE (*,450) TS(2), PS(2)
    ELSE
      WRITE (*,460)
    ENDIF
C
C   IF ENTERED CONCENTRATION IS FOR STEAM, CHL, A- & B-CRISTOBALITE
C   TEMPERATURES ARE NOT CALCULATED
C
    IF (NS .NE. 3) THEN
      WRITE (*,490)
      IF (NA(3) .NE. 1) THEN
        WRITE (*,390) TS(3)
      ELSE
        WRITE (*,400)
      ENDIF
      IF (NA(4) .NE. 1) THEN
        WRITE (*,410) TS(4)
      ELSE
        WRITE (*,420)
      ENDIF
    ENDIF

```

```

ENDIF
IF (NA(5) .NE. 1) THEN
  WRITE (*,430) TS(5)
ELSE
  WRITE (*,440)
ENDIF
ENDIF
GOTO 19

```

C

C FORMATS FOR CASE 2 OUTPUT

C

```

360 FORMAT (//2X,'density =',E11.4,' kg/m**3'//)
370 FORMAT (//20X,'Molal',11X,'Molar',12X,'ppm',8X'H4SiO4 density'
  . /14X,'mol(SiO2)/kg(f) mol(SiO2)/l(f) mg(SiO2)/kg(f) ',
  . 'kg(H4SiO4)/m**3(f)'//1X,'concentration',2X,E11.4,3(5X,E11.4)
  . ///30X,'temperature C',9X,'pressure Pa'//)
375 FORMAT (//20X,'Molal',11X,'Molar',12X,'ppm',8X'H4SiO4 density'
  . /14X,'mol(SiO2)/kg(f) mol(SiO2)/l(f) mg(SiO2)/kg(f) ',
  . 'kg(H4SiO4)/m**3(f)'//1X,'concentration',2X,E11.4,21X,E11.4
  . ///30X,'temperature C',9X,'pressure Pa'//)
380 FORMAT (8X,'quartz',6X,2(10X,E11.4))
500 FORMAT (8X,'quartz',11X,'n/a',18X,'n/a')
390 FORMAT (8X,'chalcedony',8X,E11.4)
400 FORMAT (8X,'chalcedony',13X,'n/a')
410 FORMAT (8X,'a-cristobalite',8X,E11.4)
420 FORMAT (8X,'a-cristobalite',13X,'n/a')
430 FORMAT (8X,'b-cristobalite',8X,E11.4)
440 FORMAT (8X,'b-cristobalite',13X,'n/a')
450 FORMAT (8X,'amorphous silica',6X,E11.4,10X,E11.4)
460 FORMAT (8X,'amorphous silica',11X,'n/a',18X,'n/a')
490 FORMAT (//30X,'temperature C'//)

```

C

C END OF CASE 2

C

C

C IF GIVEN OR PREDICTED TEMPERATURES OR PRESSURES
C ARE NOT WITHIN APPLICABLE RANGES...

C

```

18 WRITE (*,970)
970 FORMAT (1X/' Temp, press n/a for any solubility'//)

```

C

C PROMPT USER FOR ANOTHER LOOP, OR EXIT

C

```

19 WRITE (*,980)
980 FORMAT (/2X,'another? (y/n)')
READ (*,990) ANS
990 FORMAT (A1)
IF ((ANS .EQ. 'y') .OR. (ANS .EQ. 'Y')) GOTC 1
STOP
END

```

Subroutine H2ODR

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

 SUBROUTINE H2ODR(TSP,PSP,ROSP,ALFASP,BETASP,PSISP,USP,CVSP,CPSP,
 . HSP,GSP,SSP,VISCSP,NOP,ERRSP)

C

C DRIVER PROGRAM FOR CALCULATING THERMODYNAMIC PROPERTIES OF WATER
C FROM REYNOLDS (1979)

C

C NOP DETERMINES THE TWO INPUT PARAMETERS. TRIAL VALUES
C FOR T AND V MUST ALWAYS BE PROVIDED.

C

C IF NOP = 1, ENTER WITH T, RO
C IF NOP = 2, ENTER WITH T, P AND TRIAL RO
C IF NOP = 3, ENTER WITH P, RO AND TRIAL T
C IF NOP = 4, ENTER WITH RO, H AND TRIAL T
C IF NOP = 5, ENTER WITH T, H AND TRIAL RO
C IF NOP = 6, ENTER WITH S, RO AND TRIAL T
C IF NOP = 7, ENTER WITH S, T AND TRIAL RO
C IF NOP = 8, ENTER WITH S, P AND TRIAL T, RO
C IF NOP = 9, ENTER WITH H, P AND TRIAL T, RO
C IF NOP = 10, ENTER WITH S, H AND TRIAL T, RO

C

C PARAMETER ERR CONTROLS ACCURACY
C OF PRESSURE, ENTHALPY AND ENTROPY ITERATIONS.

C

C EXTERNAL SUBROUTINES: H201, H202

C

C CALLED BY H20, SATH20, SILSOL

C

C THIS PROGRAM USES TWO SUBROUTINES FOR THE EQ. OF STATE:

C

 (1) H201 CALCULATES THE P-RO-T VALUES;

C

 (2) H202 CALCULATES THE REMAINING PROPERTIES.

C

C THIS DIVISION CONSIDERABLY SPEEDS EXECUTION TIME FOR NOP = 2, 3

C

 IMPLICIT REAL*8 (A-H,O-Z)

 EXTERNAL H201, H202

 REAL*4 TSP,PSP,ROSP,ALFASP,BETASP,PSISP,USP,CVSP,CPSP,HSP,GSP,
 . SSP,VISCSP,ERRSP

 DIMENSION A1(7),A2(7),A3(7),A4(7),A5(2),A6(2),A7(2),A8(2),A9(7)
 . ,A10(7),ROAJ(2),TAAJ(2),BO(5),B1(6),B2(4),B3(7),B4(1),B5(1)
 . ,B(4),C(8),XM(7),DMDRO(7)

 COMMON /PROP/ A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,B0,B1,

 . B2,B3,B4,B5,B,C,ROAJ,TC,VC,PC,X0,TAAJ,R,E,XM,DMDRO

 DATA DFERR,RR/1.0E-05,461.51/

C

 ERR = ERRSP

 IF ((ERR .LE. 0.0) .OR. (ERR .GE. 1.0E-02)) ERR = DFERR

 T = TSP

 P = PSP

 RO = ROSP

 ALFA = ALFASP

 BETA = BETASP

```

PSI = PSISP
U = USP
CV = CVSP
CP = CPSP
H = HSP
G = GSP
S = SSP
VISC = VISCSP
T = T + 273.15D0
V = 1.0D0/RO
DT = 0.D0
KBR = 0
DVBF = 1.0D0
VMIN = 0.0D0
VMAX = 1.0D30
PMIN = 1.0D30
PMAX = 0.D0
DVS1 = 2.0D0*VC
DVS2 = 0.70D0*VC
KTR = 1

```

```

C
C LOOP POINT
C
1  RT = RR*T
   CALL H201(T,PX,V)
   IF (NOP .LT. 4) GOTO 5
   CALL H202(T,PX,V,ALFA,BETA,PSI,U,CV,CP,HX,G,SX,VISC)
C
C TEST FOR CONVERGENCE
C
5  GOTO (10,20,20,40,40,60,60,80,90,100), NOP
10 GOTO 700
20 IF (DABS(P-PX) .LT. (ERR*P)) GOTO 700
   GOTO 104
40 IF (DABS(H-HX) .LT. (ERR*RT)) GOTO 700
   GOTO 104
60 IF (DABS(S-SX) .LT. (ERR*RR)) GOTO 700
   GOTO 104
80 IF ((DABS(S-SX) .LT. (ERR*RR)) .AND. (DABS(P-PX) .LT. (ERR*P)))
   GOTO 700
   GOTO 104
90 IF ((DABS(H-HX) .LT. (ERR*RT)) .AND. (DABS(P-PX) .LT. (ERR*P)))
   GOTO 700
   GOTO 104
100 IF ((DABS(H-HX) .LT. (ERR*RT)) .AND. (DABS(S-SX) .LT. (ERR*RR)))
   GOTO 700
   GOTO 104
104 IF (KTR .GT. 20) GOTO 850
C
C CALCULATE THE NECESSARY PARTIAL DERIVATIVES
C
   IF (PX .LT. 0.D0) GOTO 300
   GOTO (880,120,110,120,110,120,110,110,110), NOP
C

```

C PERTURB T

C

```

110  DT = 0.001D0*T
      T1 = T + DT
      V1 = V
      CALL H201(T1,P1,V1)
      IF (NOP .LT. 4) GOTO 115
      CALL H202(T1,P1,V1,ALFA,BETA,PSI,U,CV,CP,H1,G,S1,VISC)
115  GOTO (880,880,140,140,880,140,120,120,120), NOP

```

C

C PERTURB V

C

```

120  DV = 0.001D0*V
      IF (V .LE. VC) DV = -DV
      V2 = V + DV
      T2 = T
      CALL H201(T2,P2,V2)
      IF (NOP .LT. 4) GOTO 140
      CALL H202(T2,P2,V2,ALFA,BETA,PSI,U,CV,CP,H2,G,S2,VISC)
140  GOTO (880,220,230,240,250,260,270,280,290,296), NOP
220  DPDV = (P2-PX)/DV
      IF (DPDV .GT. 0.D0) GOTO 300

```

C

C THE POINT IS GOOD- UPDATE LIMITS

C

```

      IF ((PX .GT. P) .AND. (V .GT. VMIN)) VMIN = V
      IF ((PX .LT. P) .AND. (V .LT. VMAX)) VMAX = V
      IF (V .EQ. VMIN) PMIN = PX
      IF (V .EQ. VMAX) PMAX = PX
      IF (VMIN .GE. VMAX) GOTO 840
      IF ((VMIN .GT. 0.D0) .AND. (VMAX .LT. 1.0D30)) KBR = 1
      DVBF = 1.0D0
      IF (DPDV .EQ. 0.D0) GOTO 226
      DV = (P-PX)/DPDV
      DT = 0.0D0
      GOTO 400

```

C

C DPDV = 0 AT A GOOD POINT-TREAT BY BRACKETING

C

```

226  DVBF = 0.5D0
      GOTO 300
230  DPDT = (P1-PX)/DT
      DT = (P-PX)/DPDT
      DV = 0.0DC
      GOTO 400
240  DHDT = (H1-HX)/DT
      DT = (H-HX)/DHDT
      DV = 0.0D0
      GOTO 400
250  DHDV = (H2-HX)/DV
      DV = (H-HX)/DHDV
      DT = 0.0D0
      GOTO 400
260  DSDT = (S1-SX)/DT

```

```

DT = (S-SX)/DSDT
DV = 0.0D0
GOTO 400
270 DSDV = (S2-SX)/DV
    DV = (S-SX)/DSDV
    DT = 0.0D0
    GOTO 400
280 DSDT = (S1-SX)/DT
    DSDV = (S2-SX)/DV
    DPDT = (P1-PX)/DT
    DPDV = (P2-PX)/DV
    DET = DSDT*DPDV - DPDT*DSDV
    DT = ((S-SX)*DPDV-(P-PX)*DSDV)/DET
    DV = (DSDT*(P-PX)-DPDT*(S-SX))/DET
    GOTO 400
290 DHDT = (H1-HX)/DT
    DHDV = (H2-HX)/DV
    DPDT = (P1-PX)/DT
    DPDV = (P2-PX)/DV
    DET = DHDT*DPDV - DPDT*DHDV
    DT = ((H-HX)*DPDV-(P-PX)*DHDV)/DET
    DV = (DHDT*(P-PX)-DPDT*(H-HX))/DET
    GOTO 400
296 DHDT = (H1-HX)/DT
    DHDV = (H2-HX)/DV
    DSDT = (S1-SX)/DT
    DSDV = (S2-SX)/DV
    DET = DHDT*DSDV - DSDT*DHDV
    DT = ((H-HX)*DSDV-(S-SX)*DHDV)/DET
    DV = ((S-SX)*DHDT-DSDT*(H-HX))/DET
    GOTO 400

```

C

C SPECIAL TREATMENT FOR NOP = 2, DESIGNED TO AVOID BAD ROOTS

C

300 IF (KBR .EQ. 0) GOTO 320

C

C CALCULATE SLOPE FROM BRACKETING VALUES

C

```

DPDV = (PMAX-PMIN)/(VMAX-VMIN)
V = VMAX
PX = PMAX
DV = DVBF*(P-PX)/DPDV
DT = 0.0D0
DVBF = 0.5D0*DVBFF
GOTO 400

```

C

C NOT YET BRACKETED-- ALTER V TO SEEK GOOD POINT

C

```

320 IF (V .LE. VC) DV = -0.5D0*V
    IF (V .GT. VC) DV = 0.2D0*V
    IF (VMIN .GT. 0.0D0) DV = 0.2D0*V
    IF (VMAX .LT. 1.0D30) DV = -0.5D0*V
    GOTO 400

```

C

C REGULATE THE MAXIMUM CHANGE

C

```

400  DVM = 0.2D0*V
      IF (V .LT. DVS1) DVM = 0.5D0*DVM
      IF (V .LT. DVS2) DVM = 0.5D0*DVM
      DTM = 0.1D0*T
      IF (NOP .NE. 2) GOTO 440

```

C

C SPECIAL PRECAUTIONS FOR NOP = 2

C

```

      IF (KBR .EQ. 0) GOTO 440
      VT = V + DV
      IF ((VT .GE. VMIN) .AND. (VT .LE. VMAX)) GOTO 440

```

C

C BRACKETING LIMITATION

C

```

      DV = VMIN + (P-PMIN)*(VMAX-VMIN)/(PMAX-PMIN) - V
440  DVA = DABS(DV)
      DTA = DABS(DT)
      IF (DVA .GT. DVM) DV = DV*DVM/DVA
      IF (DTA .GT. DTM) DT = DT*DTM/DTA
      T = T + DT
      V = V + DV
      KTR = KTR + 1
      GOTO 1

```

C

C NORMAL RETURN

C

```

700  IF (NOP .GT. 3) GOTO 705
      CALL H2O2(T,PX,V,ALFA,BETA,PSI,U,CV,CP,HX,G,SX,VISC)
705  T = T - 273.15D0
      RO = 1.0D0/V
      GOTO (710,720,740,760,780,790,796), NOP
710  P = PX
      S = SX
      H = HX
      GOTO 800
720  H = HX
      S = SX
      GOTO 800
740  P = PX
      S = SX
      GOTO 800
760  P = PX
      H = HX
      GOTO 800
780  H = HX
      GOTO 800
790  S = SX
      GOTO 800
796  P = PX
      GOTO 800

```

C

```

800  TSP = T

```

```

PSP = P
ROSP = RO
ALFASP = ALFA
BETASP = BETA
PSISP = PSI
USP = U
CVSP = CV
CPSP = CP
HSP = H
GSP = G
SSP = S
VISCSP = VISC
RETURN

```

C

C ERROR WRITES

C

```

840 WRITE (*,842) T, P, V, VMIN, VMAX
842 FORMAT ('H2ODR error - T,P,V,VMIN,VMAX = ',5D15.5)
RETURN
880 WRITE (*,882)
882 FORMAT ('program error in H2ODR')
RETURN
850 WRITE (*,852) NOP, T, P, V, H, S, PX, HX, SX
852 FORMAT ('H2ODR not convergent for nop = ',I3/
.      1H,7X,'T',14X,'P',14X,'V',14X,'H',14X,'S',14X,'PX',13X,
.      'HX',13X,'SX'/1H,8E15.5)
RETURN
END

```

Subroutine H201

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

SUBROUTINE H201(T,P,V)

C

C EQ. OF STATE FROM KEENAN ET AL (1978)

C

C ONLY CALCULATES PRESSURE AND SOME PARAMETERS REQUIRED

C BY H202, WHICH CALCULATES REMAINING PROPERTIES

C I/O IN MKS; CALCULATIONS IN CGS

C

C CALLED BY H20DR

C

IMPLICIT REAL*8 (A-H,O-Z)

DIMENSION A1(7),A2(7),A3(7),A4(7),A5(2),A6(2),A7(2),A8(2),A9(7)

. ,A10(7),ROAJ(2),TAAJ(2),B0(5),B1(6),B2(4),B3(7),B4(1),B5(1)

. ,B(4),C(8),XM(7),DMDRO(7)

COMMON /PROP/ A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,B0,B1,

. B2,B3,B4,B5,B,C,ROAJ,TC,VC,PC,X0,TAAJ,R,E,XM,DMDRO

C

TAC = 1.0D3/TC

RO = 1.0D-3/V

EXER = DEXP(-E*RO)

TA = 1.0D3/T

TA2C = TA - TAC

TA2AJ = TA - 2.5D0

C

C Q IS DEFINED IN EQ. 3 ON P. 128 OF KEENAN ET AL

C XM IS THE TERM IN SQUARE BRACKETS IN THAT EQ.

C

Q = 0.0D0

DQDRO = 0.0D0

DO 1 J = 1, 2

RO2AJ = RO - ROAJ(J)

XM(J) = EXER*(A9(J)+A10(J)*RO) +

. ((((((A8(J)*RO2AJ+A7(J))*RO2AJ+A6(J))

. *RO2AJ+A5(J))*RO2AJ+A4(J))*RO2AJ+A3(J))

. *RO2AJ+A2(J))*RO2AJ + A1(J)

DMDRO(J) = EXER*(A10(J)-E*(A9(J)+A10(J)*RO)) +

. ((((((7.0D0*A8(J)*RO2AJ+6.0D0*A7(J))*RO2AJ+5.0D0*A6(J))

. *RO2AJ+4.0D0*A5(J))*RO2AJ+3.0D0*A4(J))*RO2AJ+2.0D0*A3(J))

. *RO2AJ + A2(J)

IF (J .EQ. 1) TJ = 1.0D0

IF (J .EQ. 2) TJ = TA2C

Q = Q + XM(J)*TJ

DQDRO = DQDRO + DMDRO(J)*TJ

1

CONTINUE

DO 2 J = 3, 7

XM(J) = EXER*(A9(J)+A10(J)*RO) + A1(J) +

. (((A4(J)*RO2AJ+A3(J))*RO2AJ+A2(J))*RO2AJ

DMDRO(J) = EXER*(A10(J)-E*(A9(J)+A10(J)*RO)) +

. (3.0D0*A4(J)*RO2AJ+2.0D0*A3(J))*RO2AJ + A2(J)

TJ = TA2AJ*TJ

$$Q = Q + XM(J)*TJ$$
$$DQDRO = DQDRO + DMDRO(J)*TJ$$

2 CONTINUE

C

C PRESSURE IS CALCULATED FROM EQ. 11 ON P. 130 OF KEENAN ET AL
C AND THEN CONVERTED TO MKS UNITS

C

$$P = RO*R*T*(1.0DO+RO*Q+RO*RO*DQDRO)*1.0D6$$

RETURN

END

Subroutine H202

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

SUBROUTINE H202(T,P,V,ALFA,BETA,PSI,U,CV,CP,H,G,S,VISC)

C

C THERMODYNAMIC PROPERTIES OF WATER = F(RO, T); KEENAN ET AL

C THIS SUBROUTINE HAS I/O IN MKS, DEG K; COMPUTATONS ARE IN CGS, K

C

C CALLED BY H20DR; H201 MUST BE CALLED PRIOR TO CALL TO H202

C

IMPLICIT REAL*8 (A-H,O-Z)

DIMENSION A1(7),A2(7),A3(7),A4(7),A5(2),A6(2),A7(2),A8(2),A9(7)

. ,A10(7),ROAJ(2),TAAJ(2),B0(5),B1(6),B2(4),B3(7),B4(1),B5(1)

. ,B(4),C(8),XM(7),DMDRO(7)

COMMON /PROP/ A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,B0,B1,

. B2,B3,B4,B5,B,C,ROAJ,TC,VC,PC,X0,TAAJ,R,E,XM,DMDRO

C

P = P*1.0D-06

TAC = 1.0D3/TC

RO = 1.0D-3/V

EXER = DEXP(-E*RO)

TA = 1.0D3/T

DTADT = -TA/T

TA2C = TA - TAC

TA2AJ = TA - 2.5D0

C

C Q IS DEFINED IN EQ. 3 ON P. 128 OF KEENAN ET AL.

C XM IS THE TERM IN SQUARE BRACKETS IN THAT EQ.

C

Q = 0.0D0

DQDRO = 0.0D0

DQDRO2 = 0.0D0

DQDT2 = 0.0D0

TJ1 = 1.0D0

TJ2 = 1.0D0

TA2AJ1 = 0.0D0

TA2AJ2 = 0.0D0

DO 1 J = 1, 2

RO2AJ = RO - ROAJ(J)

DMDRO2 = E*EXER*(E*A9(J)+(RO*E-2.0D0)*A10(J)) +

. (((((42.0D0*A8(J)*RO2AJ+30.0D0*A7(J))*RO2AJ
+20.0D0*A6(J))*RO2AJ+12.0D0*A5(J))*RO2AJ
+6.0D0*A4(J))*RO2AJ + 2.0D0*A3(J)

IF (J .EQ. 1) TJ = 1.0D0

IF (J .EQ. 2) TJ = TA2C

Q = Q + XM(J)*TJ

DQDRO = DQDRO + DMDRO(J)*TJ

DQDRO2 = DQDRO2 + DMDRO2*TJ

1

CONTINUE

DQDTA = XM(2)

DQDTRO = DMDRO(2)

DO 2 J = 3, 7

DMDRO2 = E*EXER*(E*A9(J)+(RO*E-2.0D0)*A10(J)) +

```

        6.0D0*A4(J)*R02AJ + 2.0D0*A3(J)
DTA2AJ = (TA2AJ+(J-2)*TA2C)*TJ1
TA2AJ1 = 2.0D0*(J-2)*TJ1
IF (J .GT. 3) TA2AJ2 = (J-3)*(J-2)*TA2C*TJ2
TJ = TJ*TA2AJ
TJ1 = TA2AJ*TJ1
IF (J .GT. 3) TJ2 = TA2AJ*TJ2
Q = Q + XM(J)*TJ
DQDRO = DQDRO + DMDRO(J)*TJ
DQDRO2 = DQDRO2 + DMDRO2*TJ
DQDTA = DQDTA + DTA2AJ*XM(J)
DQDT2 = DQDT2 + (TA2AJ1+TA2AJ2)*XM(J)
DQDTRO = DQDTRO + DTA2AJ*DMDRO(J)
2    CONTINUE
C
C    DERIVATIVES OF PRESSURES AND RELATED PROPERTIES
C    SEE EQ. 11, P. 130 OF KEENAN
C
    DPDT = P/T + RO*RO*R*T*(DQDTA+RO*DQDTRO)*DTADT
    DPDRO = P/RO + RO*R*T*(Q+RO*(3.0D0*DQDRO+RO*DQDRO2))
    ALFA = DPDT/(DPDRO*RO)
    BETA = 1.0D0/(DPDRO*RO)
C
C    HELMHOLTZ FREE ENERGY (EQ. 1), INTERNAL ENERGY (EQ. 12),
C    HEAT CAPACITY AT CONSTANT VOLUME (EQ. 9), AND PRESSURE (EQ. 10)
C    ENTHALPY (EQ. 14), GIBBS FREE ENERGY (EQ. 8), AND
C    ENTROPY (EQ. 13) — ALL FROM KEENAN ET AL, PP. 128, 130
C
    XLNT = DLOG(T)
    PSI = (C(7)+C(8)/TA)*XLNT + R*T*(DLOG(RO)+RO*Q) +
          (((C(6)/TA+C(5))/TA+C(4))/TA+C(3))/TA+C(2))/TA + C(1)
    U = 1.0D3*R*RO*DQDTA + C(7)*(XLNT-1.0D0) - C(8)/TA + C(1) -
          (((4.0D0*C(6)/TA+3.0D0*C(5))/TA+2.0D0*C(4))/TA+C(3))/(TA*TA)
    CV = DTADT*((C(8)/TA-C(7))/TA+1.0D3*R*RO*DQDT2+(((2.0D1*
          C(6)/TA+1.2D1*C(5))/TA+6.0D0*C(4))/TA+2.0D0*C(3))/(TA*TA*TA))
    CP = CV + DPDT*T*ALFA/RO
    H = U + P/RO
    G = PSI + P/RO
    S = (-R)*(DLOG(RO)+RO*(Q-TA*DQDTA)) - (C(8)/1.0D3)*(1.0D0+XLNT) -
          (1.0D0/T)*((((5.0D0*C(6)/TA+4.0D0*C(5))/TA+3.0D0*C(4))/TA+
          2.0D0*C(3))/TA+C(2))/TA+C(7))
C
C    DYNAMIC VISCOSITY — FROM WATSON ET AL, 1980
C
    TS = T/TC
    TSI = 1.0D0/TS - 1.0D0
    ROS = VC/V
    ROSI = ROS - 1.0D0
    VISC = 0.0D0
    RSI = 1.0D0
    VISCO = 1.0D-6*(DSQRT(TS)/(((B(4)/TS+B(3))/TS+B(2))/TS+B(1)))
    VISC = B0(1) + TSI*(B1(1)+TSI*TSI*TSI*(B4(1)+TSI*B5(1)))
    DO 4 J = 2, 4
        RSI = RSI*ROSI

```

```

      VISC = VISC + (B0(J)+TSI*(B1(J)+TSI*(B2(J)+TSI*B3(J))))*RSI
4  CONTINUE
      VISC = VISC + ROSI*RSI*((B3(5)+B3(7)*ROSI*ROSI)*TSI*TSI*TSI
        + (B0(5)+B1(6)*TSI*ROSI))
      VISC = VISC0*DEXP(ROS*VISC)
      IF ((T .LT. 645.30D0) .OR. (T .GT. 652.55D0) .OR.
        . (RO .LT. 0.240D0) .OR. (RO .GT. 0.410)) GOTO 5
      XT = VC*VC*PC*BETA/(V*V)
      IF (XT .GT. X0) VISC = 0.922D0*VISC*XT**0.0263D0

```

C

C CONVERT FROM CGS TO MKS

C

```

5  P = 1.0D6*P
    BETA = 1.0D-6*BETA
    PSI = 1.0D3*PSI
    U = 1.0D3*U
    CV = 1.0D3*CV
    CP = 1.0D3*CP
    H = 1.0D3*H
    G = 1.0D3*G
    S = 1.0D3*S
    RETURN
    END

```

C

BLOCK DATA

IMPLICIT REAL*8 (A-H,O-Z)

DIMENSION A1(7),A2(7),A3(7),A4(7),A5(2),A6(2),A7(2),A8(2),A9(7),

. A10(7),ROAJ(2),TAAJ(2),B0(5),B1(6),B2(4),B3(7),B4(1),B5(1),

. B(4),C(8),XM(7),DMDRO(7)

COMMON /PROP/ A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,B0,B1,

. B2,B3,B4,B5,B,C,ROAJ,TC,VC,PC,X0,TAAJ,R,E,XM,DMDRO

. DATA A1/ 2.9492937D+1,-5.1985860D+0, 6.8335354D+0,

. -1.5641040D-1,-6.3972405D+0,-3.9661401D+0,-6.9048554D-1/

. DATA A2/-1.3213917D+2, 7.7779182D+0,-2.6149751D+1,

. -7.2546108D-1, 2.6409282D+1, 1.5453061D+1, 2.7407416D+0/

. DATA A3/ 2.7464632D+2,-3.3301902D+1, 6.5326396D+1,

. -9.2734289D+0,-4.7740374D+1,-2.9142470D+1,-5.1028070D+0/

. DATA A4/-3.6093828D+2,-1.6254622D+1,-2.6181978D+1,

. 4.3125840D+0, 5.6323130D+1, 2.9568796D+1, 3.9636085D+0/

. DATA A5/ 3.4218431D+2,-1.7731074D+2/

. DATA A6/-2.4450042D+2, 1.2748742D+2/

. DATA A7/ 1.5518535D+2, 1.3746153D+2/

. DATA A8/ 5.9728487D+0, 1.5597836D+2/

. DATA A9/-4.1030848D+2, 3.3731180D+2,-1.3746618D+2,

. 6.7874983D+0, 1.3687317D+2, 7.9847970D+1, 1.3041253D+1/

. DATA A10/-4.1605860D+2,-2.0988866D+2,-7.3396848D+2,

. 1.0401717D+1, 6.4581880D+2, 3.9917570D+2, 7.1531353D+1/

. DATA C/1857.065D0,3229.12D0,-419.465D0,36.6649D0,-20.5516D0,

. 4.85233D0,46.0D0,-1011.249D0/

. DATA B0/0.5132047D0,0.2151778D0,-0.2818107D0,0.1778064D0,

. -0.0417661D0/

. DATA B1/0.3205656D0,0.7317883D0,-1.070786D0,0.460504D0,0.0D0,

. -0.01578386D0/

. DATA B2/0.0D0,1.241044D0,-1.263184D0,0.2340379D0/

DATA B3/0.0D0,1.476783D0,0.0D0,-0.4924179D0,0.1600435D0,0.0D0,
-0.003629481D0/

DATA B4/-0.7782567D0/

DATA B5/0.1885447D0/

DATA B/0.0181583D0,0.0177624D0,0.0105287D0,-0.0036744D0/

DATA ROAJ/0.634D0,1.0D0/

DATA TC,VC,PC,XO/647.286D0,3.154574D-3,22.089D6,21.9D0/

DATA TAAJ/1.544912D0,2.5D0/

DATA R,E/0.46151D0,4.8D0/

END

Subroutine SATDR

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

SUBROUTINE SATDR(TS,PS,DPDTS,NOS)

C

C DRIVER FOR FINDING SATURATION CONDITIONS FROM REYNOLDS (1979)

C

C FOR NOS = 1, FINDS PSAT(T) AND (DP/DT)SAT

C FOR NOS = 2, FINDS TSAT(P) AND (DP/DT)SAT; TRIAL T IS PROVIDED

C

C INTERNAL PARAMETER ERR CONTROLS THE ITERATION ACCURACY.

C

C CALLED BY SATH20

C

IMPLICIT REAL*4 (A-H,O-Z)

EXTERNAL SATTP

DATA TC, PC /647.286E0, 22.089E6/

C

TS = TS + 273.15E0

GOTO (1, 2), NOS

C

C / SPECIFIED TSAT

C

1 IF (TS .GT. TC) GOTO 70

CALL SATTP(TS,PS,DPDTS)

TS = TS - 273.15E0

RETURN

C

C SPECIFIED PSAT - TRIAL TSAT IS FROM D. TABOR, "GASES, LIQUIDS AND

C SOLIDS" PENGUIN BOOKS, 1969, P. 208; ALONG THE ENTIRE CLAPEYRON

C CURVE, IT HAS A MAXIMUM ERROR OF 0.11 MP, AT THE CRIT. POINT.

C

2 IF (PS .GT. PC) GOTO 74

KTR = 0

ERR = 1.0E-6*PS

TS = (-4.71E3)/ALOG(PS/3.23E10)

10 IF (TS .GT. TC) TS = TC

CALL SATTP(TS,PX,DPDTS)

DP = PS - PX

IF (ABS(DP) .LT. ERR) GOTO 20

IF (KTR .GT. 20) GOTO 80

DT = DP/DPDTS

DTA = ABS(DT)

DTM = 0.10E0*TS

IF (DTA .GT. DTM) DT = DT*DTM/DTA

TS = TS + DT

KTR = KTR + 1

GOTO 10

20 TS = TS - 273.15E0

RETURN

C

C ERROR WRITES

C

```
70  WRITE (*,92) TS  
    RETURN  
74  WRITE (*,94) PS  
    RETURN  
80  WRITE (*,90) TS,PS,DPDTS,PX  
    RETURN  
90  FORMAT ('SAT not convergent for Ts, Ps, dPdTs, Px = ',4D15.5)  
92  FORMAT ('sat called for Ts = ',F6.1,' >Tc; garbage return')  
94  FORMAT ('sat called for Ps = ',1PD12.4,' >Pc; garbage return')  
    END
```

Subroutine SATTP

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

SUBROUTINE SATTP(TS,PS,DPDTS)

C

C TSAT, PSAT AND (DP/DT)SAT FOR WATER FROM KEENAN ET AL (1978)

C

C CALLED BY SATDR

C

IMPLICIT REAL*4 (A-H,O-Z)

DIMENSION F(8)

DATA F/-7.4192420E2,-2.9721000E1,-1.1552860E1,-8.6856350E-1,

1.0940980E-1,4.3999300E-1,2.5206580E-1,5.2186840E-2/

DATA TC, PC /647.286E0, 22.089E6/

C

TCS = TS - 273.15E0

TT = 0.65E0 - 0.01E0*TCS

TCC = 374.136E0 - TCS

TA = 1.0E3/TS

G = ((((((F(8)*TT+F(7))*TT+F(6))*TT+F(5))*TT+F(4))*TT+

F(3))*TT+F(2))*TT + F(1)

DGDTS = ((((((7.0E0*F(8)*TT+6.0E0*F(7))*TT+5.0E0*F(6))*TT+

4.0E0*F(5))*TT+3.0E0*F(4))*TT+2.0E0*F(3))*TT+

F(2))*(-0.01E0)

H = TA*1.0E-5*TCC*G

DHDTS = 1.0E-5*TA*(TCC*DGDTS-G*(1.0E0+TCC/TS))

PS = PC*EXP(H)

DPDTS = PS*DHDTS

RETURN

END

Subroutine SATROW

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

SUBROUTINE SATROW(TS,RHOL)

C

C DENSITY OF SATURATED LIQUID WATER FROM REYNOLDS (1979)

C

C CALLED BY SATH20, SILSOL

C

IMPLICIT REAL*4 (A-H,O-Z)

DIMENSION D(8)

DATA D/3.6711257E0,-2.8512396E1,2.2265240E2,-8.8243852E2,

2.0002765E3,-2.6122557E3,1.8297674E3,-5.3350520E2/

DATA TC,ROC/647.286E0,317.0E0/

C

TS = TS + 273.15E0

TT = (1.0E0-TS/TC)**(1.0E0/3.0E0)

X = ((((((D(8)*TT+D(7))*TT+D(6))*TT+D(5))*TT+D(4))*TT+
D(3))*TT+D(2))*TT+D(1))*TT

RHOL = (1.0E0+X)*ROC

TS = TS - 273.15E0

RETURN

END

Subroutine SATROS

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

SUBROUTINE SATROS(PS,ROV)

C

C ASSUMES LINEAR FUNCTIONS BETWEEN PRESSURE AND DENSITY

C OF STEAM PHASE OVER 5 INTERVALS

C PROVIDES A FIRST GUESS OF STEAM DENSITY FOR SATH20

C

C CALLED BY SATH20, SILSOL

C

IMPLICIT REAL*4 (A-H,O-Z)

DATA TC,PC,ROC/647.286E0,220.88E5,317.0E0/

DATA A,A1,B1,A2,B2,A3,B3,A4,B4/5.5475E-06,8.246E-06,-2.699E01,

. 1.494E-05,-1.274E02,4.3444E-05,-6.97489E02,1.768E-04,-3.60464E03/

C

IF (PS .LT. 10.0E06) ROV = A*PS

IF ((PS .GE. 10.0E06) .AND. (PS .LT. 15.0E06)) ROV = A1*PS + B1

IF ((PS .GE. 15.0E06) .AND. (PS .LT. 20.0E06)) ROV = A2*PS + B2

IF ((PS .GE. 20.0E06) .AND. (PS .LT. 21.8E06)) ROV = A3*PS + B3

IF (PS .GE. 21.8E06) ROV = A4*PS + B4

RETURN

END

Subroutine SILICA

C 2345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

SUBROUTINE SILICA(T,RHO,ALFA,RK,QTZ,AMS,CHL,ACR,BCR,NS)

C

C SILICA: CALCULATION OF KINETIC PARAMETERS AND SILICA SOLUBILITIES

C

C EXPRESSIONS FROM FOURNIER (1981, 1984),

C

FOURNIER & POTTER (1982A), RIMSTIDT & BARNES (1980)

C

C SUBROUTINE SILICA MUST BE GIVEN A TEMPERATURE, FLUID DENSITY AND
C THERMAL EXPANSION COEFFICIENT.

C

C DENSITY (RHO), THERM. EXP. COEFF. (ALFA) ARE VECTORS OF LENGTH TWO:

C RHO(1), ALFA(1) REFER TO LIQUID PHASE

C RHO(2), ALFA(2) REFER TO STEAM PHASE

C

C KINETIC RATE CONSTANT RK IS A VECTOR OF LENGTH TWO:

C RK(1) [=] KG/M*M*S

C RK(2) [=] M/S SO THAT RK(2) = RK(1)/RHO

C

C SOLUBILITIES OF CHALCEDONY, AND ALFA- AND BETA-CRISTOBALITE ARE
C VECTORS OF LENGTH 8

C SOLUBILITIES OF QUARTZ AND AMORPHOUS SILICA ARE 2 X 8 MATRICES;

C QTZ(1, 1-8) AND AMS(1, 1-8) REFER TO LIQUID PHASE

C QTZ(2, 1-8) AND AMS(2, 1-8) REFER TO STEAM PHASE

C

C INDICES 1-4 ARE FOR SOLUBILITIES:

C CHL(1), ACR(1), BCR(1),

C QTZ(1-2, 1), AMS(1-2, 1) [=] MOLAL MOLES(SIO2)/KG(H2O)

C CHL(2), ACR(2), BCR(2),

C QTZ(1-2, 2), AMS(1-2, 2) [=] MOLAR MOLES(SIO2)/LITER(H2O)

C CHL(3), ACR(3), BCR(3),

C QTZ(1-2, 3), AMS(1-2, 3) [=] PPM MG(SIO2)/KG(H2O)

C CHL(4), ACR(4), BCR(4),

C QTZ(1-2, 4), AMS(1-2, 4) [=] DENSITY KG(H4SIO4)/M*M*M(H2O)

C

C INDICES 4-8 ARE TEMPERATURE DERIVATIVES OF SOLUBILITIES:

C UNITS CORRESPOND TO INDICES 1-4 DIVIDED BY TEMPERATURE

C (EG. MOLAR/T, ETC)

C BECAUSE SOLUBILITY EQUATIONS FOR CHALCEDONY, AND ALFA- AND BETA-

C CRISTABOLITE ARE FUNCTIONS OF TEMPERATURE ONLY, THESE ARE COMPLETE
C DERIVATIVES.

C BECAUSE SOLUBILITY EQUATIONS FOR QUARTZ AND AMORPHOUS SILICA ARE

C FUNCTIONS OF TEMPERATURE AND PRESSURE, THESE ARE PARTIAL DERIVATIVES

C CALCULATED AT CONSTANT PRESSURE.

C

IMPLICIT REAL*4 (A-H,L,O-Z)

DIMENSION RK(2), QTZ(2,8), AMS(2,8), CHL(8), ACR(8), BCR(8),
RHO(2), ALFA(2)

DATA AWS,AWH,SH,LN10/6.0084E+04,9.6115E-02,1.5998E-06,2.303E/

DATA RK1,RK2/-0.707E0,-2.598E03/

DATA QA1,QA2,QA3,QA4,QA5,QB1,QB2,QC1/-4.66206E0,3.4063E-03,

```

. 2179.7E0,-1.1292E06,1.3546E08,-1.4180E-03,-806.97E0,3.9465E-04/
DATA AM1,AM2,AM3,AM4,AM5,AM6/4.278E-1,-1.22936E3,8.887E04,
. 1.1533E3,-1.2143E-2,1.2213E-5/
DATA QS1,QS2,QS3,QS4,QS5/-4.2198E01,2.8831E-01,-3.6686E-04,
. 3.1665E-07,7.7034E+01/
DATA SA1,SA2,CH1,CH2/0.26E0,7.31E02,4.69E0,1.032E03/
DATA AC1,AC2,BC1,BC2/4.78E0,1.0E03,4.51E0,7.81E02/

C
TK = T + 273.15E0

C
C RATE CONSTANT
C
RK(1) = 10.OE0**(RK1+RK2/TK)
RK(2) = RK(1)/RHO(1)

C
C NS = 1 IS FOR ONE PHASE ONLY
C NS = 2 IS FOR COEXISTING WATER AND STEAM
C
DO 1 J =1, NS
SV = 1.OE03/RHO(J)
LSV = ALOG10(SV)

C
C QUARTZ SOLUBILITY
C
A = ((QA5/TK + QA4)/TK + QA3)/TK + QA2*TK + QA1
B = QB1*TK + QB2/TK
QTZ(J,1) = 10.OE0**((QC1*TK*LSV+B)*LSV+A)
QTZ(J,2) = RHO(J)*QTZ(J,1)
QTZ(J,3) = AWS*QTZ(J,1)
QTZ(J,4) = AWH*QTZ(J,2)

C
C DERIVATIVE OF QTZ SOLUBILITY W.R.T. TEMPERATURE, AT CONSTANT PRESSURE
C (DS/DT)P = (DS/DT)RHO + (DS/DV)T * (DV/DT)P
C
DQDTV = LN10*((QC1*LSV+QB1)*LSV+QA2-(LSV*QB2+QA3+(2.E0*QA4+
3.E0*QA5/TK)/TK)/(TK*TK))

C
C (DS/DV)T*(DV/DT)P PER MASS H2O & PER VOLUME H2O, RESPECTIVELY
C
DQVM = ALFA(J)*(TK*(QB1+2.OE0*QC1*LSV)+QB2/TK)
DQVV = ALFA(J)*(TK*(QB1+2.OE0*QC1*LSV)+QB2/TK-1.OE0)
QTZ(J,5) = QTZ(J,1)*(DQDTV+DQVM)
QTZ(J,6) = QTZ(J,2)*(DQDTV+DQVV)
QTZ(J,7) = AWS*QTZ(J,5)
QTZ(J,8) = AWH*QTZ(J,6)

C
C AMORPHOUS SILICA SOLUBILITY
C
AS1 = AM1 + (AM2 + AM4*LSV + AM3/TK)/TK +
(AM5 + AM6*TK)*TK*LSV
AMS(J,1) = 10.OE0**AS1
AMS(J,2) = AMS(J,1)*RHO(J)
AMS(J,3) = AWS*AMS(J,1)
AMS(J,4) = AWH*AMS(J,2)

```

C
C DERIVATIVE OF AM. SILICA SOLUBILITY W.R.T. TEMPERATURE,
C AT CONSTANT PRESSURE; SEE QUARTZ ABOVE
C

$$\text{DASDTV} = \text{LN10} * ((\text{AM5} + 2.0\text{EO} * \text{AM6} * \text{TK}) * \text{LSV} - ((\text{AM2} + \text{AM4} * \text{LSV}) + 2.0\text{EO} * \text{AM3} / \text{TK}) / (\text{TK} * \text{TK}))$$

$$\text{DASVM} = \text{ALFA}(\text{J}) * (\text{TK} * (\text{AM5} + \text{AM6} * \text{TK}) + \text{AM4} / \text{TK})$$

$$\text{DASVV} = \text{ALFA}(\text{J}) * (\text{TK} * (\text{AM5} + \text{AM6} * \text{TK}) + \text{AM4} / \text{TK} - 1.0\text{EO})$$

$$\text{AMS}(\text{J}, 5) = \text{AMS}(\text{J}, 1) * (\text{DASDTV} + \text{DASVM})$$

$$\text{AMS}(\text{J}, 6) = \text{AMS}(\text{J}, 2) * (\text{DASDTV} + \text{DASVV})$$

$$\text{AMS}(\text{J}, 7) = \text{AWS} * \text{AMS}(\text{J}, 5)$$

$$\text{AMS}(\text{J}, 8) = \text{AWH} * \text{AMS}(\text{J}, 6)$$

1 CONTINUE

C
C CHALCEDONY SOLUBILITY
C

$$\text{CHL}(3) = 10.0\text{EO} * (\text{CH1} - \text{CH2} / \text{TK})$$

$$\text{CHL}(1) = \text{CHL}(3) / \text{AWS}$$

$$\text{CHL}(2) = \text{CHL}(1) * \text{RHO}(1)$$

$$\text{CHL}(4) = \text{RHO}(1) * \text{SH} * \text{CHL}(3)$$

C
C DERIVATIVE OF CHALCEDONY SOLUBILITY W.R.T. TEMPERATURE
C

$$\text{CHL}(7) = \text{CHL}(3) * \text{LN10} * (\text{CH2} / (\text{TK} * \text{TK}))$$

$$\text{CHL}(5) = \text{CHL}(7) / \text{AWS}$$

$$\text{CHL}(6) = \text{CHL}(5) * \text{RHO}(1)$$

$$\text{CHL}(8) = \text{RHO}(1) * \text{SH} * \text{CHL}(7)$$

C
C ALPHA-CRISTOBALITE SOLUBILITY
C

$$\text{ACR}(3) = 10.0\text{EO} * (\text{AC1} - \text{AC2} / \text{TK})$$

$$\text{ACR}(1) = \text{ACR}(3) / \text{AWS}$$

$$\text{ACR}(2) = \text{ACR}(1) * \text{RHO}(1)$$

$$\text{ACR}(4) = \text{RHO}(1) * \text{SH} * \text{ACR}(3)$$

C
C DERIVATIVE OF ALPHA-CRISTOBALITE SOLUBILITY W.R.T. TEMPERATURE
C

$$\text{ACR}(7) = \text{ACR}(3) * \text{LN10} * (\text{AC2} / (\text{TK} * \text{TK}))$$

$$\text{ACR}(5) = \text{ACR}(7) / \text{AWS}$$

$$\text{ACR}(6) = \text{ACR}(5) * \text{RHO}(1)$$

$$\text{ACR}(8) = \text{RHO}(1) * \text{SH} * \text{ACR}(7)$$

C
C BETA-CRISTOBALITE SOLUBILITY
C

$$\text{BCR}(3) = 10.0\text{EO} * (\text{BC1} - \text{BC2} / \text{TK})$$

$$\text{BCR}(1) = \text{BCR}(3) / \text{AWS}$$

$$\text{BCR}(2) = \text{BCR}(1) * \text{RHO}(1)$$

$$\text{BCR}(4) = \text{RHO}(1) * \text{SH} * \text{BCR}(3)$$

C
C DERIVATIVE OF BETA-CRISTOBALITE SOLUBILITY W.R.T. TEMPERATURE
C

$$\text{BCR}(7) = \text{BCR}(3) * \text{LN10} * (\text{BC2} / (\text{TK} * \text{TK}))$$

$$\text{BCR}(5) = \text{BCR}(7) / \text{AWS}$$

$$\text{BCR}(6) = \text{BCR}(5) * \text{RHO}(1)$$

BCR(8) = RHO(1)*SH*BCR(7)

C

RETURN

END

Subroutine TSI02

C 345678-1-2345678-2-2345678-3-2345678-4-2345678-5-2345678-6-2345678-7-2

C

 SUBROUTINE TSI02(TS,RHO,SOLT,NS)

C

C TSI02: CALCULATION OF EQUILIBRIUM TEMPERATURES FROM CONCENTRATION

C

C SOLUBILITY EXPRESSIONS FROM FOURNIER (1981,1984),

C

 FOURNIER & POTTER (1982A, B)

C

C EQUATIONS FOR QUARTZ AND AMORPHOUS SILICA ARE AVAILABLE AS FUNCTIONS
C OF TEMPERATURE, OR TEMPERATURE AND DENSITY; THE FORMER IS USED AS
C A FIRST BEFORE USING THE LATTER

C

C EQUATION FOR CHALCEDONY, AND ALFA- AND BETA-CRISTABOLITE ARE WHOLLY
C TEMPERATURE DEPENDENT

C

C TEMPERATURES (TS) ARE CALCULATED FROM:

C

C (1) CONCENTRATION (SOLT) IN MOLAL UNITS

C

C (2) DENSITY (RHO) WHERE RHO IS A VECTOR OF LENGTH TWO;

C

 RHO(1) = DENSITY FOR LIQUID PHASE OF COEXISTING LIQUID AND
C STEAM IN EQUILIBRIUM WITH QUARTZ

C

 RHO(2) = DENSITY FOR LIQUID PHASE OF COEXISTING LIQUID AND
C STEAM IN EQUILIBRIUM WITH AMORPHOUS SILICA

C

C (3) OPTION PARAMETER (NS);

C

 1 = USE THE FULL EQUATIONS

C

 2 = USE EQUATIONS THAT ARE ONLY FUNCTIONS OF TEMPERATURE

C

 3 = STEAM-PHASE, FULL EQUATIONS: CO-EXISTING WATER & STEAM

C

 4 = WATER-PHASE, FULL EQUATIONS: CO-EXISTING WATER AND STEAM

C

C CONCENTRATIONS ARE STORED IN A VECTOR: SOL(1) - MOLAR OF SIO2

C

 (2) - MOLAL OF SIO2

C

C NOTE THAT UNITS 1 AND 4

 (3) - PPM OF SIO2

C

C REQUIRE A KNOWN FLUID DENSITY

 (4) - KG(H4SIO4)/M*M*M(H2O)

C

C EQUILIBRIUM TEMPERATURES ARE STORED IN A VECTOR: TS(1) - QUARTZ

C

 (2) - AM. SILICA

C

C NOTE THAT THE LAST THREE MINERALS

 (3) - CHALCEDONY

C

C HAVE WHOLLY TEMPERATURE-DEPENDANT

 (4) - ALFA-CRIST

C

C SOLUBILITY EQUATIONS ONLY

 (5) - BETA-CRIST

C

 IMPLICIT REAL*4 (A-H,L,O-Z)

 DIMENSION TS(6), RHO(2)

 DATA AWS,AWH,SH,LN10 / 6.0084E4, 9.6115E-2, 1.5998E-6, 2.303E0/

 DATA RK1,RK2 /-0.707E0,-2.598E3/

 DATA QA1,QA2,QA3,QA4,QA5,QB1,QB2,QC1 /-4.66206E0, 3.4063E-3,

 . 2179.7E0,-1.1292E6, 1.3546E8,-1.4180E-3,-806.97E0, 3.9465E-4/

 DATA AM1,AM2,AM3,AM4,AM5,AM6 / 4.278E-1,-1.22936E3, 8.887E4,

 . 1.1533E3,-1.2143E-2, 1.2213E-5/

 DATA QS1,QS2,QS3,QS4,QS5 /-4.2198E1, 2.8831E-1,-3.6686E-4,

 . 3.1665E-7, 7.7034E+1/

 DATA SA1,SA2,CH1,CH2 / 4.52E0, 7.31E2, 4.69E0, 1.032E3/

 DATA AC1,AC2,BC1,BC2 / 4.78E0, 1.0E3, 4.51E0, 7.81E2/

```

C
SV = 1.OE03/RHO(1)
LSV = ALOG10(SV)
SOLP = SOLT*AWS

C
C FOR A CRUDE GUESS OF TEMPERATURE OF LIQUID WATER COEXISTING WITH
C STEAM, USE THE WHOLLY TEMPERATURE-DEPENDENT EQUATIONS
C
IF (NS .EQ. 2) GOTO 1

C
C CALCULATE TEMPERATURE FROM QUARTIC SOLUTION OF QUARTZ SOLUBILITY
C EQUATION GIVEN BY FOURNIER & POTTER (1982A)
C
AAA = QA2 + LSV*(LSV*QC1+QB1)
A1 = (QA1-ALOG10(SOLT))/AAA
A2 = (QA3+QB2*LSV)/AAA
A3 = QA4/AAA
A4 = QA5/AAA
A = A1*A3 - A2*A2/3.OE0 - 4.OE0*A4
B = A2*(A1*A3-2.OE0*A2*A2/9.OE0+8.OE0*A4)/3.OE0
  - A1*A1*A4 - A3*A3
XX = SQRT(B*B/4.OE0+A*A*A/27.OE0)
AA = (-B/2.OE0+XX)**(1.OE0/3.OE0)
FB = -B/2.OE0 - XX
BB = (ABS(FB))**(1.OE0/3.OE0)
IF (FB .LT. 0.0) BB = -BB
Y = A2/3.OE0 + AA + BB
R = SQRT(A1*A1/4.OE0-A2+Y)
XDE = 3.OE0*A1*A1/4.OE0 - 2.OE0*A2
Q = (4.OE0*A1*A2-8.OE0*A3-A1*A1*A1)/(4.OE0*R) - R*R
D = (SQRT(XDE+Q))/2.OE0

C
C ONLY ONE ROOT IS NEEDED FOR THE QUARTZ SOLUTION
C
TS(1) = -A1/4.OE0 + R/2.OE0 + D - 273.15EO

C
C IF CONCENTRATIONS ARE LOW ENOUGH, WHOLLY THE TEMPERATURE DEPENDENT
C SOLUTION IS ADEQUATE FOR AMORPHOUS SILICA IN LIQUID PHASE
C
IF (((NS .EQ. 1) .OR. (NS .EQ. 4))
  .AND. (SOLT .LT. 1.6E-2)) GOTO 3

C
C CALCULATE TEMPERATURE FROM QUARTIC SOLUTION OF AMORPHOUS SILICA
C SOLUBILITY EQ. IN TEXT (FOURNIER, 1984)
C
IF (NS .EQ. 4) LSV = ALOG10(1.OE3/RHO(2))
BBB = AM6*LSV
B1 = AM5*LSV/BBB
B2 = (AM1-ALOG10(SOLT))/BBB
B3 = (AM2+LSV*AM4)/BBB
B4 = AM3/BBB
A = B1*B3 - B2*B2/3.OE0 - 4.OE0*B4
B = B2*(B1*B3-2.OE0*B2*B2/9.OE0+8.OE0*B4)/3.OE0
  - B1*B1*B4 - B3*B3

```

```

XX = SQRT(B*B/4.OE0+A*A*A/27.OE0)
FA = -B/2.OE0 + XX
AA = (ABS(FA))**(1.OE0/3.OE0)
FB = -B/2.O - XX
BB = (ABS(FB))**(1.OE0/3.OE0)
IF (FA .LT. 0.OE0) AA = -AA
IF (FB .LT. 0.OE0) BB = -BB
Y = B2/3.OE0 + AA + BB
R = SQRT(B1*B1/4.OE0-B2+Y)
XDE = 3.OE0*B1*B1/4.OE0 - 2.OE0*B2
Q = (4.OE0*B1*B2-8.OE0*B3-B1*B1*B1)/(4.OE0*R) - R*R
D = (SQRT(XDE+Q))/2.OE0

```

```

C
C ONE OF 2 ROOTS MAY BE CORRECT FOR AMORPHOUS-SILICA SOLUTION
C

```

```

IF ((NS .EQ. 3) .AND. ((TS(2) .LT. 250E0) .OR.
      (TS(2) .GT. 374.136E0))) THEN
  TS(2) = -B1/4.OE0 + R/2.OE0 - D - 273.15E0
ELSE
  TS(2) = -B1/4.OE0 + R/2.OE0 + D - 273.15E0
ENDIF
GOTO 2

```

```

C
C CALCULATE TEMPERATURE FROM LIQUID WATER SOLUBILITY EQ. FOR QUARTZ
C ALONG THE WATER-STEAM CURVE, FROM FOURNIER & POTTER (1982B).
C THIS EQ. IS A FUNCTION OF TEMPERATURE ONLY
C

```

```

1 TS(1) = QS1 + SOLP*(QS2+SOLP*(QS3+SOLP*QS4)) + QS5*ALOG10(SOLP)

```

```

C
C CALCULATE TEMPERATURES FOR LOW PRESSURE, LIQUID WATER SOLUBILITY
C EQUATIONS FOR AM. SILICA, CHALCEDONY, ALPHA- & BETA-CRISTOBALITE
C GIVEN IN FOURNIER (1981)
C

```

```

3 TS(2) = SA2/(SA1-ALOG10(SOLP)) - 273.15E0
2 TS(3) = CH2/(CH1-ALOG10(SOLP)) - 273.15E0
  TS(4) = AC2/(AC1-ALOG10(SOLP)) - 273.15E0
  TS(5) = BC2/(BC1-ALOG10(SOLP)) - 273.15E0
  RETURN
END

```